An Interop Publication

# ConneXions®

## The Interoperability Report

May 1992 — *Fifth Anniversary Issue* — Volume 6, No. 5

*ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.*

## In this issue:



*ConneXions* is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP.
E-mail: connexions@interop.com.



ISSN 0894–5926

## From the Editor

Welcome to INTEROP 92 Spring and to the Fifth Anniversary Issue of *ConneXions—The Interoperability Report*. This edition contains articles directly and indirectly related to the conference and tutorial program. We realize that with so much going on you probably won't have much time to read during INTEROP week, but take this copy with you and read it later. And don't forget to subscribe at the special conference discount rate.

When I look back over the last five years it becomes clear that a great deal has happened in the area of internetworking and interoperability. The TCP/IP technology has become mature, stable and "off-the-shelf." You'll find several articles in this issue that describe how people are using their internetworks.


One word, more than any other, describes what has been and is happening: *Growth*. The global Internet now has several million users, over 700,000 hosts, and so many individual networks that part of the address space is rapidly becoming exhausted.

The growth in all aspects of networking is also reflected in what Interop Company does: Our ever-expanding attendance figures have made it necessary to host two INTEROPs per year, and *ConneXions* continues to grow in average page count, there is just *so* much to cover!

I normally use this page to introduce the articles, but since this is an anniversary issue, I'm going to break the rules and let the articles speak for themselves. Instead, I want to acknowledge all those people who have made *ConneXions* possible for the last 5 years. First and foremost on my list are the over 175 authors whose valuable contributions make my editing and publishing job a joy. I also want to thank Jim Matthews and all the folks at Globe Printing Company for putting up with all my questions and teaching me that "printing is a craft, not a science." Special thanks to Dwight Hare for help with Macintosh and UNIX, Mike St. Johns for *PostScript* clues, and Mary Salman for the nice masthead design. Last, but not least, thanks to Bonnie Cantoni, the *ConneXions* coordinator, for her tireless work with the database and for making sure that all the subscribers get their copy every month.

I'm also going to break another rule this month by publishing an opinion piece of my own. Since we're hosting "The Great OSI Debate" on Thursday evening, I decided to throw in my two cent's worth on this controversial topic. —*Ole J. Jacobsen, Editor and Publisher*

# Components of OSI: Inter-Domain Routing Protocol (IDRP)

by Susan Hares, Merit

## Introduction

Out of the fertile ground of the Internet and the traditional ANSI subcommittees has sprung an OSI *Inter-Domain Routing Protocol* (IDRP). This protocol is officially called:

"Information Processing Systems—Telecommunications and Information Exchange between Systems—Protocol for Exchange of Inter-Domain Routeing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs (CD 10747)."

This article provides an overview of the IDRP protocol in terms of its architecture and features. No attempt has been made to describe the format of packets or state machines. These details are better left to the protocol document. Charlie Kunzinger (IBM), the editor of the ISO IDRP document, has kept this document readable, and deserves a lot of thanks for his efforts.

IDRP has progressed along the standards track to *Committee Draft* (CD) Ballot. CD Ballot is the second stage in the ISO standards process. This stage is similar to "Proposed Internet Standard" status in the IP Internet world. The current protocol description is firm foundation for implementation but, like all standards at the Committee Draft level, IDRP is still open to possible technical changes.

Merit demonstrated a prototype implementation of IDRP during INTEROP 91 Fall. Experience with the prototype has been fed back into the IDRP specification. For additional information about the IDRP prototype, please contact Merit (e-mail address: `nsfnet-info@merit.edu`).

Special thanks go to Charlie Kunzinger (IBM), the editor of IDRP, whose ten pages of overheads were turned into this document. Thanks are also due to Dave Katz (Merit) and Yakov Rekhter (IBM) who reviewed this document extensively.

## Architectural overview

The OSI routing environment is described in ISO TR 9575 [1] as a set of interconnected *Routing Domains* (RDs). RDs are groups of hosts and routers that operate according to the same routing plan and are administered by a single authority. Routing domains interact with each other in a "mutually suspicious manner" [1] due to their mutual independence and autonomy. Routing domains may each have their own view of what constitutes an optimal route. The OSI routing environment does not require that all routing domains have homogenous criteria (policy) about how to select an optimal route.

In many ways a routing domain is like an *Autonomous System* in the IP Internet [2]. Routing information passed within a routing domain (*Intra-Domain*) is trusted. Routing information passed between domains (*Inter-Domain*) may not be trusted. IDRP provides a means to communicate routing information and to calculate routes between routing domains. IDRP does not require that all routing domains have homogeneous criteria (policy) for route selection [3].

Routing knowledge needs to be spread throughout an internet in some ubiquitous manner if packets are going to reach their destinations. However, the independent and autonomous status of routing domains within an internet work against providing homogeneous routing information. In this environment, some routing domains (such as NSFNET) need to carry traffic on behalf of other routing domains.

Routing domains that carry traffic on behalf of other routing domains are known as *Transit Routing Domains* (regional networks and backbone networks, for example).

Since a routing domain has only a finite amount of resources, it must control how these resources are used. For example, NSF does not want to subsidize XYZ Corporation by allowing the commercial traffic from this company to go across NSFNET. To control what traffic passes through the network, NSFNET controls what routing information it accepts and transmits. By using this control, NSFNET does not carry the commercial traffic for XYZ corporation.

In an atmosphere in which each routing domain needs to control the traffic passed through it by controlling the dissemination of routing information, one RD may *filter* the routing information it receives from another RD. By filtering the routes, a routing domain keeps only the routing information needed to do its job. A routing domain may also filter what it sends to other RDs so that only certain pathways are used. In addition, configurations are set up to authenticate the remote peer sending the information. This filtering and authentication creates a "firewall" to keep each routing domain independent of other routing domains.

**Types of routing domains**

Routing domains whose policies do not permit them to carry transit traffic are known as *End Routing Domains.* End Routing Domains may be further subdivided into those connected to a single adjacent RD (Stub) and those connected to multiple adjacent RDs (multihomed). In figure 1, C is a *transit routing domain* and B is a *multihomed routing domain.* Routing domain A is a *stub routing domain.* Routing domain A is *adjacent* to routing domain C.

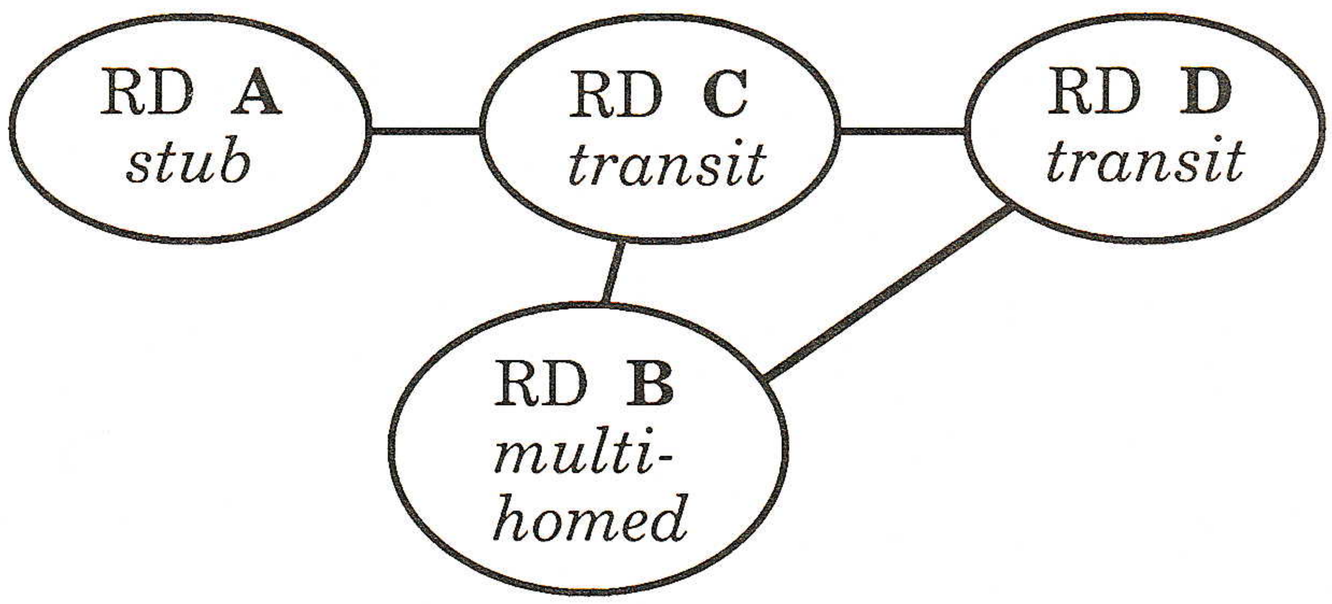



Figure 1: Routing Domains

**Inter-domain routing goals**

IDRP is designed to provide for the deterministic selection of paths through the OSI environment (internet) taken by network layer packets. Included in the design criteria for IDRP are the ability to:

- Scale well to a growing Internet, and
- Allow transit routing domains to transit traffic.

Additionally, IDRP assumes that the OSI environment (internet) may face multiple cuts to the topology of the network. Therefore IDRP provides routing that adapts to the topological changes (at the Inter-Domain level).

When a protocol scales well, it will make efficient use of network bandwidth and the processing and memory within routers.

*continued on next page*

## Inter-Domain Routing Protocol *(continued)*

IDRP was designed to fit into the OSI stack without affecting the existing network layer protocols. These protocols are:

- CLNP: Connectionless Network Layer Protocol (ISO 8473) [3]
- ES–IS: End System to Intermediate System Routeing Exchange Protocol (ISO 9452) [4]
- IS–IS: Intermediate System to Intermediate System Intra-Domain Routeing Protocol (ISO 10589) [5]

IDRP does not require either ES–IS or IS–IS in order to operate properly. The only requirement placed on the intra-domain routing environment is the presence of stable paths between border routers in the same RD participating in IDRP (for the forwarding of transit traffic) and the presence of stable paths to and from end systems within the routing domain (for the forwarding of traffic originating or terminating within the local RD). IDRP was not designed to provide user data security, to adapt routes based on traffic load, or to repair partitions of routing domains.

### Multiple Qualities of Service

The OSI connectionless network layer protocol (ISO 8473) allows a network packet to request several *qualities of service* (QOS). This QOS is similar to *types of service* (TOS) in IP. These QOS values allow a packet to be "colored." IDRP supports the calculation of separate paths for each of the different qualities of service by means of *Distinguishing Attributes* [5], thus allowing paths or routes to be "colored."

When an ISO 8473 packet comes into a router, the router will check to see if the packet has been "colored" by QOS. If so, the router looks up the destination address in the appropriate QOS's routing table. The IDRP protocol specifies a mapping between the QOS values (packet's "color") and an IDRP inter-domain pathway (the route's "color"). If no QOS value has been set, the default routing table is used.

Most of the internet will probably start out having only a "default" QOS (or colorless pathway). Packets without any QOS (colorless packets) may be able to travel much further than "colored" packets. If a pathway of the right QOS (color) is not available, the packet is dropped.

The CLNP QOS function or an IDRP QOS pathway does not have to be supported by each system in the pathway. However, each system must correctly handle a packet with a QOS value set. For example, suppose a particularly miserly end system wanted to send a packet along pathways which had minimal costs. He would set the "Expense" bit (perhaps like a green color) in the network layer packet and shove the packet out the door to a router. If IS–IS is used within a routing domain, the local routing domain would pass the packet on a low expense route until it hit a router on the border of the routing domain. This border router would check in its routing tables to find a route which had low expense. If it found one, the miserly end system's packet would speed along low cost routes. If a route was not found, the packet would be discarded and an ERROR PDU would be passed back to the miserly end system. Given that our miserly end system did not want to spend lots of money reaching the destination, this behavior makes the end system happy. If the miserly end system must spend the money to send this packet, it can switch to a route with no QOS set (colorless) when it receives the ERROR PDU.

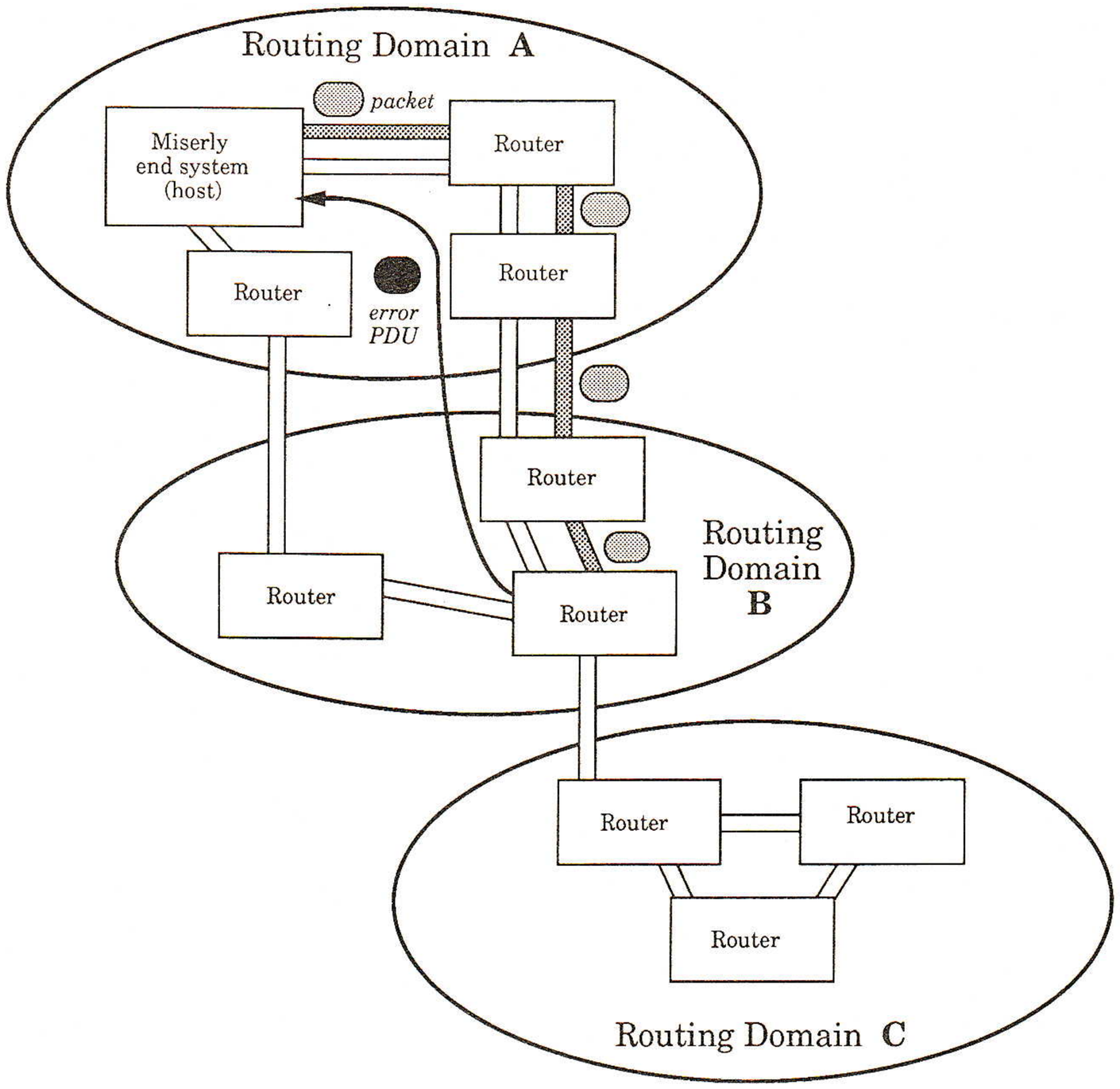



Figure 2: Miserly Packet example

**Scope of QOS**

The "color" of a packet (QOS) or route (Distinguishing Attribute) has a *scope*. The scope may either be specific to:

- A source address (a route may have scope for a group of source addresses),
- A destination address (a route may have scope for a group of destination addresses), or
- A globally unique QOS such as the Expense flag.

Two types of "color" for the packet or route (QOS and Distinguishing Attributes) deal with source addresses: Source Specific QOS and Source Specific Security QOS. Two types of "color" deal with destination addresses: Destination Specific QOS and Destination Specific Security QOS.

To indicate a globally unique QOS value, a value is set in the header of a CLNP (ISO 8473) packet in an option field. For example, the packet from our miserly end system would have a bit set that indicates "Expense" QOS. The "Expense" Distinguishing Attribute which IDRP maps to the "Expense" QOS has a value which indicates how expensive each route is. However, the meaning of that "Expense QOS" value may be valid only within an RD or a set of RDs.

Global availability or agreement on the colors of "packets" or routes will probably take a great deal of coordination in the Inter-Domain realm. Today's Internet is built on such coordination between networks. The current policies on how routes are passed in the Internet are implemented in router configurations in lots of networks.

For the scope of a "color" of a packet (QOS) or a route (Distinguishing Attribute) to truly be global, many countries may have to agree on the exact syntax and meaning of a "color."

*continued on next page*

## Inter-Domain Routing Protocol *(continued)*

For example, NSF currently limits any traffic passing across the NSFNET to being non-commercial and in support of research and education. This type of policy restriction may be illogical within Japan or Australia. If our miserly packet was from a University, it could find a "golden" pathway that was the research pathway through the US. But when it hit the US border, the definition of what this "golden" pathway means may simply not apply to Canada or Japan or Australia or imply something different within those countries.

### Protocol overview

A router that participates in inter-domain routing (and thus runs the IDRP protocol) is known as a *Border Intermediate System* (BIS). A routing domain may contain one or more BISs. A BIS uses policy to select among the routes it receives. In order for the IDRP protocol to work, all the BISs within an RD must have a reliable and consistent picture of how to route packets. To minimize the network bandwidth and processing power used by IDRP, the protocol must help control the amount of information passed. The next sections stroll through the features provided by the IDRP protocol.

IDRP is part of the suite of routing protocols for the OSI Network Layer, and runs over the OSI Connectionless Network Layer Protocol (CLNP ISO 8473). IDRP describes how PDUs are exchanged between BISs, how routes are constructed, and how protocol errors are handled, but the choice of how routes are selected is left to the local BIS's policy. BIS PDUs are passed between BISs within a domain and between BISs in different routing domains.

### Path Vector

IDRP is a "path vector" protocol, which is neither a distance vector nor a link state protocol. The distance metric of a distance vector protocol is not required. Unlike a link state protocol, the full topology and explicit link status are not distributed. In a "path vector" protocol, the routes distributed contain a complete path to the source of the route. Such a protocol provides the following:

- A vector of path attributes,
- One route to destination for each QOS value,
- Excellent information hiding/abstraction and
- Support for hop-by-hop routing.

Inclusion of path information allows more rapid convergence than classic distance vector protocols and eliminates the "count to infinity" problem.

### Information flow

IDRP operates between pairs of BISs as shown in Figure 3. Similar to the BGP protocol in the IP Internet, IDRP is used between BISs within the same routing domain so that the entire RD has a consistent picture of inter-domain routing.

IDRP allows each BIS to control routing information passed on to other BISs. Only routes which satisfy the local policy of a BIS, and have been selected to be to be used, will be passed on to the next BIS.

IDRP exchanges routing information between BISs within a domain and between adjacent BISs in adjacent routing domains. Since IDRP routing updates are incremental, routing traffic between BISs are minimized. The heaviest flow of traffic comes when a BIS first becomes operational. The BIS must establish a connection to other BISs in its RD and adjacent RDs. Routing information must be sent from the adjacent BISs to this new BIS.

A BIS's route re-calculation is partial and event-driven (only newly arrived routes need to be evaluated). The events which cause route re-calculation are:

- A BIS receives an incremental routing update with new routes,
- A BIS neighbor goes down, or
- A BIS neighbor comes up.

**Loop suppression**

The richly interconnected Internet topology provides the opportunity to construct paths that form loops. However, IDRP will never construct looping paths because it keeps track of all RDs traversed by a route. If a BIS sees its own RD in a route advertised by a BIS in another RD, it will ignore that route.

**Routing Domain Identifiers**

IDRP uses a *Routing Domain Identifier* (RDI) to uniquely identify a Routing Domain. The Routing Domain Identifier is taken from the same address space as the *Network Service Access Point* (NSAP) addresses. RDIs are drawn from the NSAP address space solely to simplify administration; no relationship should be inferred between the value of an RDI and the NSAP addresses resident within the RD.

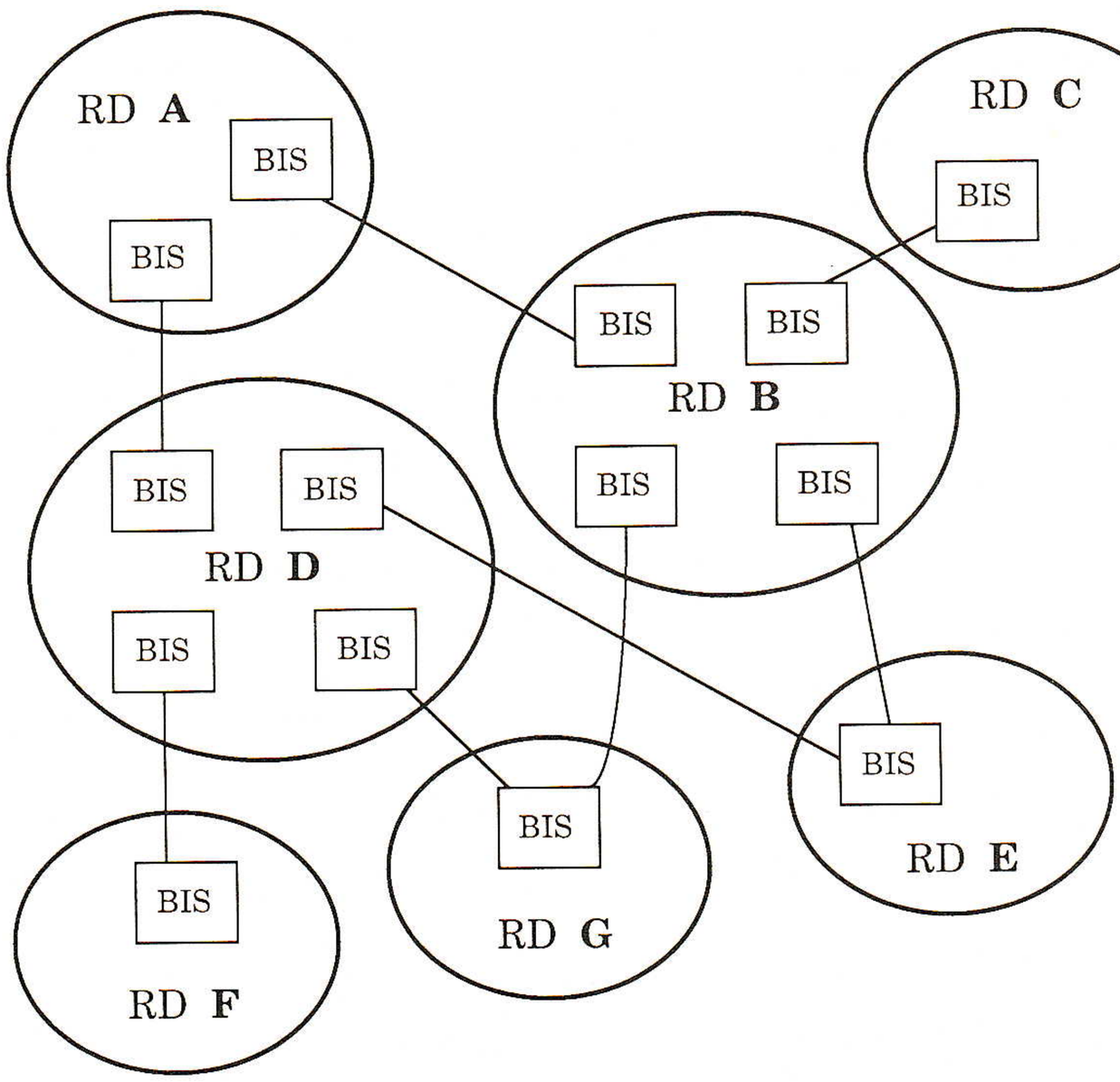



Figure 3: Routing Domains connected by BISs

**Path Attributes**

Like BGP, IDRP defines a pathway as a pairing between a destination and the attributes of that path to the destination. A destination in IDRP is described by *Network Layer Reachability Information* (NLRI). NLRI contains groups of NSAP addresses described by NSAP prefixes. IDRP describes pathways through the morass of routing domains as ordered sequences of RDIs. For example, the pathway in Figure 3 between A and E could be A, B, E.

In IDRP, the attributes of a path to a destination describe the characteristics of the path. Of these attributes, some are "Distinguishing attributes" and color the routes. Figure 4 shows which attributes can be listed together on a pathway.

*continued on next page*

## Inter-Domain Routing Protocol *(continued)*

Like your mother, when she gave you choices of vegetables, you can choose one from Group A, one from Group B and one from Group C. Unlike your mother, you also have the choice of "none," which is called "default" routing (no vegetables).

| *Group A* | *Group B* | *Group C* |
|---|---|---|
| • Transit Delay<br>• Residual Error<br>• Expense<br>• Capacity<br>• Source Sensitive QOS<br>• Destination Sensitive QOS | • Priority | • Source Sensitive Security<br>• Destination Sensitive Security |

Figure 4: Possible Groups of Distinguished attributes

**Information bases**

For each unique combination of distinguishing attributes and destinations, there is local policy that affects the router. IDRP policy is akin to BGP policy. It is policy that is set by a routing domain administrator for a routing domain and expressed in local configuration files on each node. The functioning of the "global" internet policy is the combination of all these "local" policies.

Each BIS builds a *Routing Information Base* (RIB) based on routing information received from other BISs and from within the local RD. The RIB represents the set of routes that has been selected for use by the BIS.

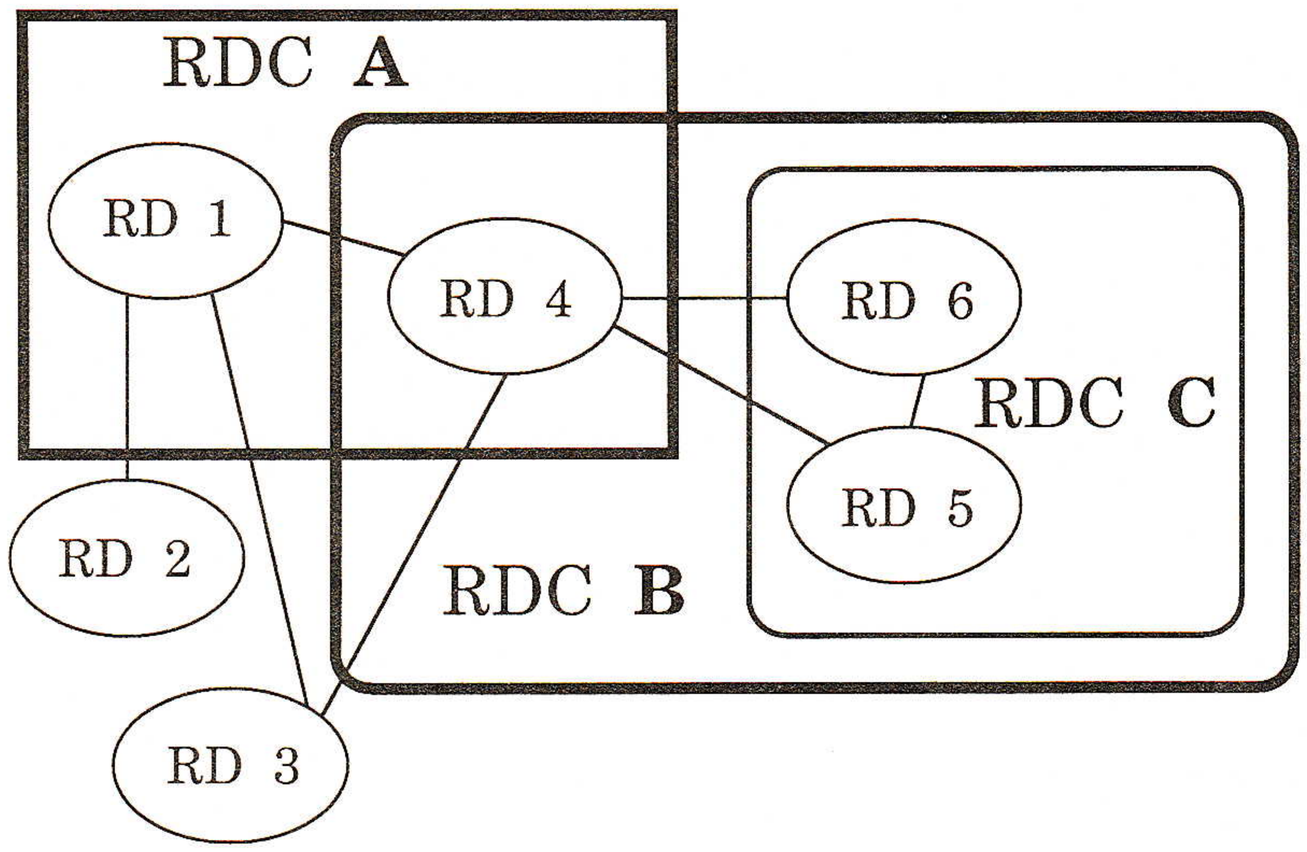
Each remote BIS neighbor sends a BIS its RIB for each unique set of distinguishing attributes. For example, if a BIS supports an expense attribute (for our miserly network user) and a route with default QOS, a Routing Information Base will be sent for expense QOS and for default QOS. This routing information contains only the active routes selected by the neighbor.

The local BIS, upon receiving this information, uses local policy to select the routes, and updates its RIB. From the RIB, the BIS generates a *Forwarding Information Base* (FIB). The FIB effectively contains a set of destinations and next hop BISs for each destination.

Upon the loss of a BIS neighbor, the local BIS will re-run the route selection function. The local BIS will use local policy to select among the routes from the remaining neighbors, update its RIB, and generate a new FIB.

**Routing Domain Confederations**

A *Routing Domain Confederation* (RDC) is a group of routing domains that join together in such a way that they appear to be a single routing domain as viewed from outside the RDC. The only common policy that must be supported among the members of the RDC is that all routes between members of the RDC must lie entirely within the RDC. RDCs provide a powerful mechanism for reducing the complexity of routing information, since the details of the internal topology of the RDC are hidden from those domains outside the RDC. In addition, the size of the RD path information carried by IDRP will be reduced. RDCs can overlap and/or be nested. Figure 5 shows 6 RD organized into 3 RDCs. RDCs A and B overlap. RDC C is nested inside of RDC B.




*6 RDs, organized into 3 Routing Confederations*

Figure 5: Routing Domain Confederations

IDRP provides good scaling by a variety of mechanisms. Three types of information can be abstracted or hidden: Network reachability information, topology information, and transit policies. Topology information is expressed in terms of RD pathways to the remote destination.

The abstractions IDRP provides are comforting in the face of an expanding internet. This abstraction of data will lessen the routing information that is passed around. Less bandwidth devoted to routing means there is more bandwidth for the internet's real purpose: sending data from here to there.

**Use of OSI NSAP structure**

Network reachability information can be abstracted or hidden by using the hierarchical nature of the NSAP address to group NSAPs under shorter NSAP prefixes. Such prefixes are carried in IDRP routes. Figure 6 shows an example of how several addresses are combined into a shorter prefix. A part of an NSAP address used to describe a group of NSAP addresses is called an *NSAP Prefix.*

```
47.0005.80.ffff00.0000.0001.0001.010203040506.01
47.0005.80.ffff00.0000.0001.0002.80ff32401f23.01
47.0005.80.ffff00.0000.0001.0003.800103040500.01
```

*could be represented by one of the following prefixes, among others (going from most specific to most general):*

`47.0005.80.ffff00.0000.0001` or `47.0005.80.ffff00` or `47.0005` or `47`

Figure 6: Combination of NSAPs into a Prefix

**Compression of RD pathway information**

In combination with the use of NSAP prefixes, RD path information can be abstracted from an ordered sequence of RDIs to an unordered set of RDIs. Although some information is lost, the automatic loop suppression mechanism in IDRP is preserved. For example, if paths to two destinations have RD paths A, B, C and A, B, D, the single unordered set A, B, C, D could be used and a single path created. This abstraction may allow multiple routes to be aggregated based on the policies of a routing domain.

*continued on next page*

## Inter-Domain Routing Protocol *(continued)*

Topology information can be abstracted using the IDRP concept of Routing Domain Confederations. RDCs provide a nice means for scaling. Fewer policies need to be administered by other routing domains. Fewer total routes need to be reported to the world outside the RDC. Inside an RDC the policies are exploded to serve the needs of RDs within the Confederation. The IDRP protocol can distinguish between intra-RDC and trans-RDC routing information. In addition, transit policies may be implemented in the presence of an RDC via the "Hierarchical Recording" feature.

Transit policies can also be abstracted by using RDCs. RDCs can be nested, allowing policy for several routing domains to be summarized as a single policy of a Routing Confederation.

### Taming the Shrew

In the play by William Shakespeare, "Taming of the Shrew," a husband seeks to tame a head-strong wife. The wife has a strong verbal flow and a stronger temper. The wife learns that ranting and raving can be more effective if it is truthful, used sparingly, and directed at the right person.

RFC 1104 describes the same sort of global taming for the Internet. The Internet, like the wife, has a strong flow of information which must be tamed by correct policies in order to let packets peacefully flow. Routes, like "ranting and raving," must be directed at just the right routing domains.

IDRP provides a taming or reduction of routing information and processing via control of the distribution of information. Only the actual routes that are selected to be used by a BIS are propagated to other routing domains. Like what "truth in advertising" should be for home appliances, the BIS only advertises what it uses. We should have the same restriction on those people who sell vacuum cleaners or coffee makers.

Each RD may also select the set of RDs to which its routes are distributed. In this way, transit routing policies are supported via control of the distribution of routing information. Again, to use the analogy of home appliances, a company could choose to not ship chain saws to someone under 16 years of age. Route selection processing is also reduced because routes are selected based on routes received (actual routes only) plus the local BIS's policy, which can further reduce the number of routes.

The IDRP attributes "DIST_LIST_INCL" and "DIST_LIST_EXCL" give routing domains the means to select the routing domains that will receive their routing information. The "DIST_LIST_INCL" allows routing information to be sent to RDIs listed in the attribute. "DIST_LIST_EXCL" allows routing information to be sent to any RDI not included in the attribute list. An RDI in one of these attributes can specify either a routing domain or a routing domain confederation.

The use of Routing Domain Confederation also helps tame the flow of routing information needed. The "Hierarchical Recording" attribute helps track when RDCs have been entered and exited so routes can be limited to within an RDC. Once an RDC has been entered and the "Hierarchical Recording" attribute is set, routes can be advertised only to BISs that can be reached without exiting any RDCs.

An RDC can be nested within an RDC or overlap another RDC. Routes can be announced to RDs:

- Within the same RDC,
- In an RDC nested inside this RDC, or
- In an overlapping RDC.

This establishes a hierarchy of RDs as you enter further into nested or overlapping RDCs and controls route transitivity. Disjoint RDCs are RDCs which can only be reached by exiting all routing confederations in which this local BIS resides. The "Hierarchical Recording" attribute cannot be passed between two Disjoint RDCs.

**Interaction with intra-domain routing**

IDRP may need to interact with intra-domain routing if it is running in an RD that contains end systems. First of all, IDRP needs to know the set of NSAP prefixes reachable within the local routing domain. This information is likely to be statically configured information, and not extracted dynamically from intra-domain routing.

Secondly, a representation of the destinations reachable outside of the RD may be injected into intra-domain routing so that the appropriate exit BIS can be reached. This information may be provided to intra-domain routing in a dynamic fashion.

Both of these interactions can be accomplished by the manipulation of managed objects via network management primitives.

**Reliability**

Like the truck ads that say "Ford tough," the IDRP protocol is built to be "Internet tough." All communications between peer BISs can be protected through the use of a cryptographic signature on a per-packet basis.

Unlike BGP, IDRP implements its own reliable transport. Each data-carrying packet (BIS protocol data unit (PDU)) contains a sequence number which is used for re-transmission and flow control. Data-carrying PDUs are the OPEN PDU (which contains the connection information), the UPDATE PDU (which contains incremental routing information), and the RIB REFRESH PDU (which contains routing information). A retransmission timer controls the retransmission of packets. The method of flow control is a window scheme. This scheme only allows a fixed number of BIS PDUs to be transmitted to a remote BIS neighbor prior to the local BIS receiving an acknowledgement.

**Especially neat features**

Thanks to the rich technical environment of the Internet, the contributors to the IDRP specification have put in several neat features to make it work even better than BGP. Two of these features which have made it into the current CD ballot version of the document are:

- A BIS playing route server for another router
- RIB Refresh PDU

Some features which are under discussion but not in the CD ballot are: Auto-configuration of BISs and Source-based routes.

The Route server function in IDRP allows one BIS to do policy and handle the BIS PDUs but point to another router to do the packet switching. Suppose your BIS was running on a slow packet pusher (perhaps a UNIX system running *gated*). With the route server you can point your data flow to a fast packet pusher of a router (say one of the commercially available routers) and traffic would not be impacted by the IDRP protocol running on a slow machine.

*continued on next page*

## Inter-Domain Routing Protocol *(continued)*

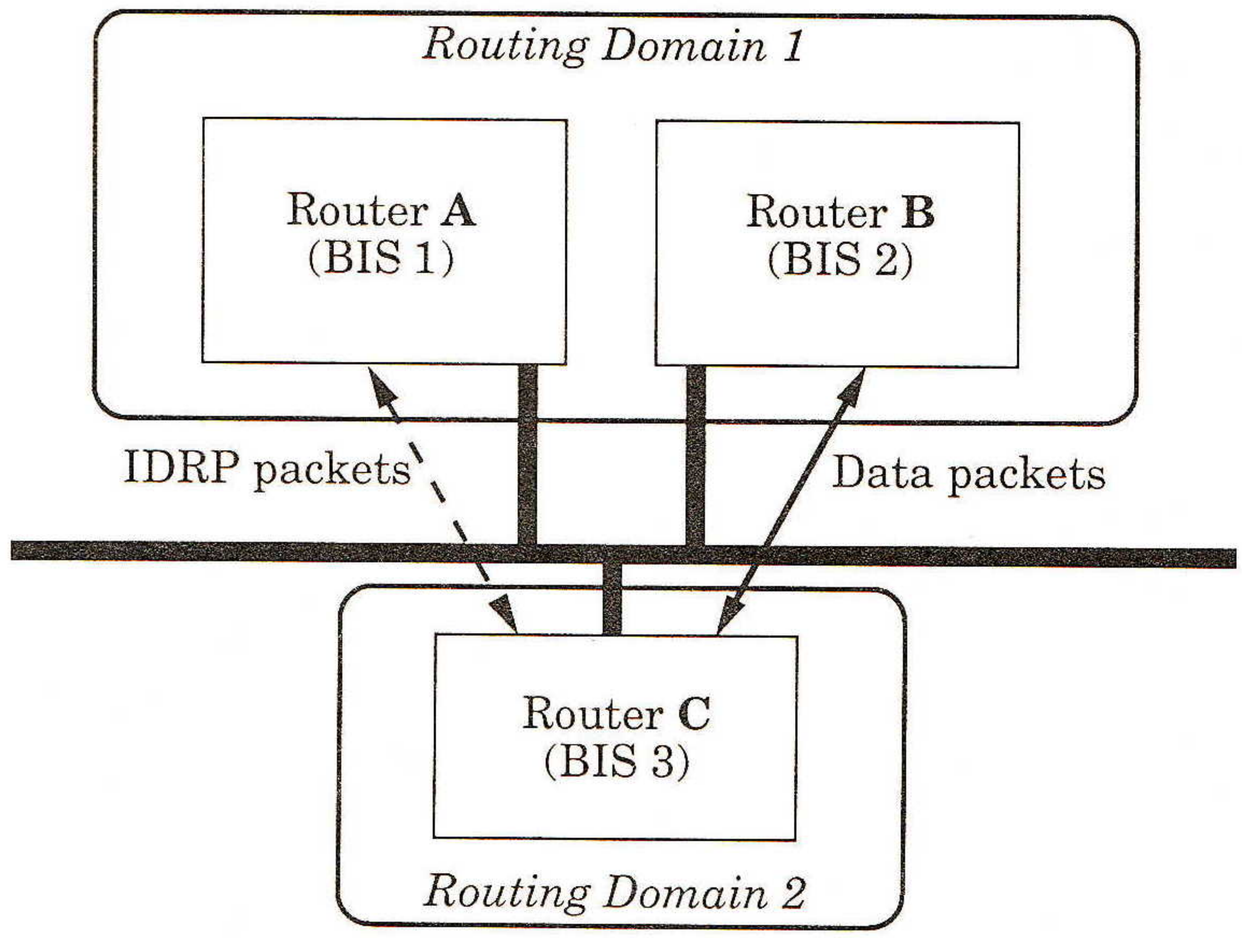



Figure 7: Route Server

By using a RIB REFRESH PDU, a BIS may refresh its Routing Information Base (RIB) from a neighbor BIS or send its active routing information to a neighbor BIS. If a router detects that its routing information has been corrupted, it can get fresh IDRP routing information from all its neighbors. Or perhaps something convinces the BIS (or the person operating the router) that a neighbor BIS has scrambled the routing information advertised by the local BIS. The local BIS simply ships the remote BIS a new copy of the RIB.

Auto-configuration of BISs involve the methods by which BISs can communicate with each other without having to pre-configure each neighbor BIS. This feature could make it easier to add new BISs to an RD without having to reconfigure all other BISs.

Each network packet (NPDU) has a source and destination field just like a good old IP packet. Source-based routing establishes routes based on both source and destination addresses. Source-based routing may help "commercial" traffic take a different route than "research" traffic or provide two different routes to the same destination, each tagged for a different set of source addresses.

### Conclusion

IDRP reminds one of "Stone Soup." "Stone Soup" is described in a children's fairy tale in which three soldiers return from a war and no one in the village wants to give them any food. They start with 3 stones in a pot of water. By the end of the tale, the villagers have added all the vegetables, meat and potatoes you could ever want.

IDRP started with the concepts of BGP and the OSI routing framework. These two original concepts are like the "stones" in the "stone soup." To this beginning, the experience and needs of the Internet community and the ANSI X3S3.3 communities have been added to make IDRP one OSI protocol full of "meat" and substance.

IDRP has already followed the tradition of Internet protocols by having a prototype developed by Merit during the CD Ballot stage. Nothing improves a protocol like an implementor. Often implementors cannot wait until they "talk" to the person that wrote a protocol specification. This dialogue is most painful when you are both the implementor and the person that helped with the original text.

Perhaps this tutorial has given you an appetite to read the protocol and any associated documents. Below you will find a reading list to help you fill up on OSI routing issues. Bon appetit!

## References


[1] "Information processing systems—Telecommunications and Information Exchange between systems—OSI Routeing Framework," (ISO TR 9575).

[2] Hares, S. & Katz, D., "Administrative Domains and Routing Domains: A Model for Routing in the Internet," RFC 1136.

[3] "Information processing systems—Telecommunications and Information Exchange between systems—Protocol for Providing the Connectionless-Mode Network Service," (ISO 8473), March 1987.

[4] "Information processing systems—Telecommunications and Information Exchange between systems—End System to Intermediate System Routeing Exchange Protocol for use in conjunction with the Protocol for providing connectionless-mode network service, (ISO 8473)," (ISO 9542), March 1988.

[5] "Information processing systems—Telecommunications and Information Exchange between systems—Intermediate system to Intermediate System Intra-Domain Routeing Exchange protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)," ISO 10589, forthcoming.


## Reading List

[A] RFC 1136, RFC 1237, RFC 1195.

[B] Tutorials on the ISO routing documents [3], [4], [5], found on `merit.edu` in directory `/pub/iso/noop/tutorial`.

[C] The actual ISO documents [1], [3], [4], [5], and IDRP. Working copies of IS–IS, and IDRP can be found on `merit.edu` for ANSI X3S3.3 committee work in the `/pub/iso` directory.

[D] ANSI X3S3.3 notes and mail archives on IDRP. The ANSI X3S3.3 archives are on `merit.edu`. The ANSI X3S3.3 mailing list is `x3s33@merit.edu`.

[E] *ConneXions,* Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[F] *ConneXions,* Volume 5, No. 1, January 1991, "Special Issue: Inter-domain Routing."


**SUSAN HARES** has her B.S.E. in Computer Engineering from the University of Michigan. She works for Merit on the NSFNET project. Susan is active in IETF and ANSI standards committees. Susan is a network warrior on her third tour of duty. Her first tour of duty was with a private network which supported X.25, lots of terminal servers, and Bisync. The network supported the "back-end" processing of the stock market sales. People relied on it for financial information to make investments. When things failed, bank officers lost out on investments and yelled at the nearest network person. Enduring this verbal abuse helped Susan learn the value of reliable networks. Her second tour of duty was in the MAP/TOP OSI world. She participated on standard committees writing the MAP/TOP 3.0 specification. She also helped develop products for MAP/TOP protocol suite. Susan learned a lot about what makes OSI succeed or fail from this tour of duty. The company Susan worked for made industrial computers. To encourage each engineer to do their best to make all network products (software and hardware) reliable, a true story was told to each Engineer. This story detailed how a computer error caused a computer controlled robot to put a bucket of molten metal through the wall of a factory. Susan's latest tour of duty has been with Merit on the NSFNET project. During the early days she worked on Routing Coordination for the TCP/IP Internet. Lately, Susan has been working on OSI in the NSFNET and the Internet. During INTEROP 91 Fall in October 91, she helped organize a very successful Internet-wide OSI demonstration between 30 companies. She can be reached as `skh@merit.edu`.

# The Internet Explorer in Japan

## by Carl Malamud

[Ed.: The following excerpt is adapted from *Exploring the Internet*, the INTEROP Book for the Fall '92 Conference. Written by Carl Malamud, the book will be published by Prentice Hall. Carl describes *Exploring the Internet* as a "technical travelogue." In six months, he has gone three times around the world, playing the digital tourist in over 60 cities in 23 countries. Over the next few months, we will run a few extracts from this book. Register for the INTEROP 92 Fall Conference and be the first to get copy of this new book].

**Tokyo**

Arriving at Tokyo's Narita airport ahead of schedule, I waited two hours for my bus into town. After an hour on the bus, we were still wending our way through the Tokyo suburbs, passing Disneyland, the docks, and threading through ever larger mazes of freeways.

Looking down, I realized our freeway was built on top of a river. Next to us, only a few feet away, were rows and rows of office buildings. Looking inside each, I could see rows and rows of desks all crowded together. Below us, but above the river, ran the railway tracks, above us another freeway.

The next morning, I ventured forth on the subway system to find the University of Tokyo, known as *Todai.* There, I met Professor Haruhisha Ishida, a professor at the Computer Centre. Professor Ishida proceeded to give me an excellent introduction to networks in Japan.

**Networks**

Japan, like the US, has many different networks. BITNET, as in most countries, started by being funded by IBM, but is now member supported. What membership fees don't cover is provided by the main sponsors, the Science University of Tokyo. Japan's BITNET has a 56Kbps link to CUNY and provides tail links to Korea and Taiwan.

Another network, *N-1net,* is an older proprietary network to tie together mainframe systems with services like remote job entry and remote login. N-1net is managed by the *National Center for Science and Information Systems* (NACSIS). NACSIS, a research institute funded by the Ministry of Education, also maintains a 56Kbps link to the National Science Foundation in Washington.

NSF uses a portion of the line to search several large databases maintained by NACSIS. A small part of the line is available for BITNET and TCP transfers and is used mostly for mail exchange from their X.400 messaging system. NACSIS hopes to upgrade the US link to 192Kbps in 1992, at which time it would become one of the key international links for Japan.

A third network is the *Todai International Science Network* (TISN, pronounced "Tyson"). TISN is used by physicists and chemists and is based on the DECnet protocol suite. TISN maintains a 128Kbps link between Todai and the University of Hawaii. Due to political walls between ministries, Japan also maintains HEPnet connectivity at the High Energy Physics Laboratory (KEK) to the Lawrence Berkeley Laboratory in California.

**WIDE**

One last network is JAIN, which is a TCP/IP-based university network that links the university LANs together. This whole plethora of networks is tied together by JUNET, based on UUCP, and WIDE (*Widely Integrated Distributed Environment*), the Japanese Internet.

There are two paths between the Japanese Internet and the rest of the Internet. WIDE maintains a 192Kbps link from Keio University in Fujisawa to the University of Hawaii. In addition, the 128Kbps link between Todai and Hawaii, used primarily for DECnet traffic, acts as an automatic backup in case the 192Kbps link has problems.

**Jun Murai**

WIDE is an interesting network in the Japanese system. All the other networks are funded by the Ministry of Education or another group. Officially, WIDE doesn't exist. Even more amazing, in the tightly segmented world of Japanese politics, commercial and educational users are all mixed together.

I asked Professor Ishida how such a situation could come to be. His answer was quite simple.

"Jun Murai."

Jun Murai used to be a research associate at Todai, working with Ishida, but recently moved south to his alma mater, Keio University. In the staid world of academics, Jun is a fairly remarkable character.

Just for starters, he wears blue jeans. Failing to dress in the regulation dark blue suit has caused no small amount of comment among senior faculty members. Jun gets away with it because he really knows what he is doing. He commands a loyal following among students, has the respect of all his peers, and has even won the grudging respect of his seniors.

The WIDE network is based on donations of money from corporations and graduate students from education. The network lives almost hand to mouth. Money is funneled into Murai's research programme and is used to pay for the network. It was not unusual at times to have the coffers get down to 1 or 2 months of operating costs, forcing Murai into perpetual fundraising.

After this introduction to Japanese networks, Professor Ishida gave me a tour of the Todai facilities he runs. The main campus is wired with three FDDI backbones, one for TCP/IP, one for DECnet, and a third for administrative computing. A fourth 400Mbps backbone is used for video. Fanning out from each of the backbones are UTP-based Ethernets. These local networks form the basis for terminal clusters, workstations, and even the supercomputers.

Professor Ishida led me past an empty room stuffed with Hitachi mainframes and supercomputers into a terminal cluster. The cluster was divided up into cubicles and was dead quiet. At the entrance was a color video display with a map of the cluster. The occupied cubicles had red dots inside, the free ones were green, allowing people to find an available workstation or terminal without disturbing the people already working.

Off to the side was a room with 9 track tape and cartridge drives. Each drive had a terminal in front of it with a menu system to help users do their own tape work.

Another room was filled with printers. When a user prints a job at Todai, it automatically spools to disk. Each printer has a card reader attached to it (credit card, not punch card, that is). A user walks up to the reader, slides an ID card through, and jobs are retrieved from disk and printed. A terminal in front of each printer indicates how long the current queue is.

*continued on next page*

## The Internet Explorer in Japan *(continued)*

Leaving the cluster, we went down to the first floor where Professor Ishida pointed to a large electronic signboard on the wall. The board displayed the current status of the mainframes, including the number of jobs and the expected delay before a new job would begin processing. No need to walk upstairs and login if the system is slow.

Walking outside, I felt a little whir under my feet as the automatic brushes on the doormat came to life, cleaning my shoes. I headed down to the subway.

Coming off the subway, I ducked into a dark noodle shop where a line of salary men all sat hunched over the counter slurping noodles. Hoping my neighbor hadn't chosen chicken lips, I pointed to his dish and was promptly handed a steaming bowl of delicious miso soup.

Feeling refreshed and refurbished, I went back to my hotel, entering at the same time as LaToya Jackson and a very large entourage. I fell in with the entourage and smiled graciously at the hotel reception committee.

On the way to the elevators, we were waylaid by a group of six American tourists all bearing labels to identify their tour group and all armed with cameras. Figuring LaToya could handle this one alone, I slipped into the elevator that three hotel staff members had been guarding for her. Before they could say anything, I punched my floor number and the startled attendants went jumping out in search of another elevator.

### Fujitsu network

I spent the afternoon in the Pub Misamu, waiting for an evening dinner engagement with Tomoo Okada, general manager of Fujitsu's Value Added Group. Worried from a prior telephone conversation that his English was only slightly better than my Japanese, I figured a few beers would get me in the mood for an evening of nodding and smiling. (It turned out that his English was actually quite good.)

My appointment with Tomoo Okada was at 7 PM. At precisely 7:01, my phone rang. I descended to the lobby to meet what was obviously a very senior manager, a distinguished-looking executive in his early 50s. Okada supervises over 700 people, including Fujitsu's Value Added Network division, two wholly owned subsidiaries, and a collection of divisions of other subsidiaries. His main responsibility is running a network which is used by both Fujitsu and customers.

The network is essentially an X.25 backbone with 144 local access nodes spread throughout Japan. 250 leased lines running at speeds from 129Kbps to 6Mbps form the network backbone. Running on the network are classic protocols such as X.28 and X.29 for asynchronous terminals, and synchronous protocols for 3270 and Fujitsu terminals. As a general rule, full protocol stacks don't run over the network, although an insurance company has begun deploying an OSI CONS-based service over X.25. International links include 64Kbps lines to Sydney, Singapore, Korea, Hong Kong, and Malaysia. Higher-speed links are provided to the UK, the US, and Germany.

### NiftyServe

Perhaps the most interesting project Okada supervises is *NiftyServe*, a wholly-owned subsidiary that acts as the licensee to the US *CompuServe*. *NiftyServe* preserves the famous *CompuServe* user interface, but the software was entirely rewritten to run on a UNIX platform and to support Kanji characters.

To make *NiftyServe* start off with a bang, 32,000 Fujitsu employees were given accounts. Over 70 percent of those accounts are active users. The reason for this high utilization is quite simple: "important meetings," such as promotion reviews, are posted here.

Some of the most devoted users are overseas Japanese employees. In addition to providing things like daily Japanese language news, the service has proved important in another respect. In Japan, when family or friends die, it is considered very important to immediately express condolences. With Fujitsu or any large corporation, of course, fellow employees *are* family. Before the bulletin board service started, it could take days for the postal service to deliver the news overseas, forcing Fujitsu employees into the unwilling position of being impolite.

*NiftyServe* has added a few other interesting twists to the classic bulletin board. For example, you can instruct the system to redirect your mail to a fax machine. You can even go to a pay phone and have your mail read to you.

All this information about Fujitsu was imparted to me with rapid-fire delivery over over a seemingly infinite parade of dishes in a dim sum restaurant. After polishing off a half-dozen large bottles of beer, Okada suggested we switch to a Chinese aged wine, similar to sake, but darker in color (and at least as potent).

Even with all these drinks, Tomoo Okada kept up a steady delivery of information on Fujitsu. With my head swimming and my stomach stuffed, I stumbled back to my hotel to await the next day's pilgrimage to Fujisawa to visit Jun Murai, the *Internet Samurai.*

**Fujisawa**

The next morning I was up early, having heard horror stories of how hard it would be navigate the succession of subways and trains in the middle of rush hour. Yes, there are packers to get more people into the subways. Since the density was about equivalent to a crowd leaving a football game, however, the packers weren't yet needed. They sat watchfully by, wearing white gloves and ready to jump into action when required to.

What was amazing was that the density continued. Passing the Tokyo suburbs, I expected the crowds to disperse, but the Yokohama commute brought even more people onto the trains. Even outside of Yokohama in Fujisawa, the density remained constant.

Arriving at Tsujido station, I waded through huge crowds until I got outside the station. There, I started waving Jun Murai's card around and mispronouncing "Keio," my destination, until a kindly old gentleman in an indeterminate uniform pointed me to the stop for bus number zero.

I queued up and boarded. We passed through Fujisawa and started to head into the countryside. Throughout the ride, everytime we passed a cluster of university-like buildings, I would accost some hapless passenger and mumble "Keio." One by one, they waved for me to sit down.

Finally, we reached the end of the line. Rising out of the countryside, sitting on a hill, was a futuristic clump of buildings, surrounded by empty fields. I made my way up a long set of stairs into the central courtyard. In marked contrast to the subways, trains, and streets, constantly teeming with people, the Fujisawa campus of Keio University, Japan's best private university, was absolutely deserted.

*continued on next page*

## The Internet Explorer in Japan *(continued)*

Feeling like I was in some futuristic ghost town, I wandered around random buildings until I chanced upon a group of administrators hidden away in a back office. Waving Jun's business card produced a flurry of whispered consultations until a map of the campus suddenly appeared and I was pointed towards building number zero.

Going to the third floor, I knew I was in the right place when I saw an office with a pile of empty computer boxes piled outside. I parked myself on the boxes and waited for Jun. It was an eerie feeling, sitting there in this deserted building on this deserted campus listening to the building creak and groan. Occasionally, a door would open and a person would scoot out of a room at the end of a hallway and scamper away, leaving only the echoes of their footsteps.

I was to find later that the deserted campus was merely an illusion. Being a self-employed idler, I had arrived well past 9, by which time everybody was hard at work.

**Junsec**

Jun Murai's secretary arrived at 9:30. Known worldwide as "Junsec," her e-mail address appears on all WIDE literature, posters, stickers, and the like. Junsec let me onto an X terminal to read the several hundred mail messages that had accumulated since Honolulu. Most of the mail was about *Bruno,* the standards server.

Evidently, word had gotten out. Bruno was running 24 hours a day, with load averages of 35 packets per second not uncommon. So many mail requests had come into the infoserver software that the batch queue had over 150 unfulfilled jobs. The amount of FTP traffic was so heavy that we had become a serious user of bandwidth on the transatlantic links.

This posed a delicate problem. On the one hand, we wanted lots of people to use the server in Colorado so that we could prove to the ITU that the service was needed. On the other hand, being good network citizens meant setting up mirrored servers.

What was interesting was that offers to provide mirrored servers were pouring in from all over the world. Comparing notes with Tony Rutkowski, it appeared that every country in Europe had volunteered to maintain a set of the standards. Several hundred hosts had already received files from Bruno and it looked like we were getting FTP traffic from over 25 countries.

When I was almost done plowing through my mail, Jun Murai came bursting in. Disheveled, animated, and wearing his trademark blue jeans, he greeted me enthusiastically. Taking me into the obligatory conference room, we began talking while Junsec brought in cups of tea (she had already brought me two cups of coffee).

I presented Jun with a copy of *STACKS* for his boss, Dean Aiso. He presented me with a handsomely bound copy of the WIDE annual report, several hundred pages of kanji with gold-embossed lettering on the cover. My manners being better than my Japanese, I thanked him while mentally trying to figure out how to get this heavy and, to me at least, cryptic tome back to the States.

Formalities duly dispensed with, we started talking about WIDE. Jun is only 36 years old and I was curious how, in a stratified society like Japan, somebody like him had ended up running the Japanese Internet.

**History**

In 1984, NTT still had a throttle on all telecommunications. Putting foreign devices (i.e., modems) on the telephone lines would be considered about as proper as greeting the Emperor by slapping him on the back. However, it was widely known that in April, 1985, NTT would deregulate.

All the senior researchers had been discussing how to take advantage of deregulation to put in networks. Meetings were held to debate the subtleties of various OSI architectures. To Jun, this was a waste of time. As he puts it, "I was young and that was boring."

He took two modems, scrammed a phone line from university administrators (no easy feat), and started running *uucp* transfers. That was the start of *JUNET*. While the establishment continued to attend OSI meetings, JUNET continued to grow. Links were set up to a machine named *mcvax* in Amsterdam (the precursor to EUnet) and to *seismo* in Washington, D.C. (the precursor to UUnet). By 1986, a domestic IP network had started and by 1989, Jun Murai and Torben Nielsen had set up links to Hawaii. A crucial element of connectivity was provided by Larry Landweber, who hooked up Japan to CSNET.

Meanwhile, the powers that be continued to debate OSI. When they finally looked up from their deliberations, Jun already had several hundred nodes on his network. As in much of the world, while committees waited for OSI, a few people turned TCP/IP networks into a reality.

The waiting is ending in many places. NACSIS has seen the light on multiprotocol networks and was beginning a TCP/IP network in mid-1992. NACSIS is funded by the Ministry of Education which requires networks with no involvement from the commercial sector. This meant that WIDE would still function as the Japanese Internet and that the NACSIS network would use access control to prohibit commercial traffic from polluting the university network. IP-level access control lists yields interesting results in true distributed applications such as AFS.

Over the years, WIDE had grown to the point where it had an annual budget of US $1 million by the 1991 school year, raised totally by donations. WIDE is officially a research project, involving 57 researchers, roughly half from the university community. The operational requirements of WIDE became so demanding that Jun was actually turning away requests to be connected until a suitable infrastructure could be set up to run the net.

Repeatedly during our conversation, Jun referred to a desire to stop running networks and get back to his real research. Even with the demands of running WIDE, he has compiled an impressive record.

**Kanji support**

When JUNET was first being put into place, Jun noticed he had a network but nobody was using it. Networks are like a fine dinner: the whole effort is wasted if nobody comes. The reason, in the case of JUNET, was quite simply that e-mail and USENET used the Roman alphabet and Japan uses *kanji* characters. Jun changed that. He added kanji support to X, kanji character handling for RFC 822 mail, and multibyte character handling for the C programming language.

He also helped design a font server. This software lets users start spelling out characters in romanized Japanese. As the words begin to be formed, Kanji characters start appearing on the screen. When the right character shows up, the user points to it and proceeds to the next word.

*continued on next page*

## The Internet Explorer in Japan *(continued)*

**Phone Shell**

When I visited Fujisawa, Jun was involved in fascinating projects to support mobile hosts and mobile people. One of the more interesting projects was the *Phone Shell*. The Phone Shell is a fully functional UNIX shell which takes input from a 12-key telephone pad instead of a keyboard or a mouse. Typically, users will execute scripts, such as having mail messages redirected into a voice synthesizer.

The Phone Shell can do any UNIX operation. As Jun explains it, "I can go to the bar and drink beer. I go to a phone and *ping* my routers, and if they are still working, I go back and drink more beer."

**ISDN**

Another project brings ISDN into the TCP/IP protocol suite. Unlike other countries, which are still demonstrating the viability of demos, Japan has deployed ISDN as an operational service. In fact, WIDE uses ISDN lines to supplement leased lines in case of congestion or failure. The WIDE ISDN module is fully integrated into TCP/IP. When a datagram for an ISDN-reachable source is encountered, a call is placed. The delay to set up the circuit ranges from 800ms to 3 seconds. Once established, the line stays up until it has been idle for a user-configurable period, at which point it is taken down.

ISDN can be used for more than just routers, however. Jun has TCP/IP on his laptop. Tokyo has ISDN pay phones. Think about it, you can bring a laptop into a phone booth and be a fully functioning member of the Internet. Even Superman would be jealous.

Another project close to operation when I visited was the use of satellite circuits for home PCs. Japan has deployed satellites as an alternative to cable TV. Dishes cost as little as $100 each. Satellites for TV signals are a one-way data channel, but they operate at high bandwidth. A PC with an ISDN card can use 64Kbps "B" channels to send commands. Data coming back to the PC can use up to the 8Mbps wide-area bandwidth of the satellite dish.

Ethernet cards are used to give the PC a 10Mbps interface in a LAN environment. Jun and his staff have linked the Ethernet card to the satellite receiver to give the home user the ability to do WAN-based multimedia, large file transfers, and other operations requiring large amounts of bandwidth.

We went from Jun's laboratories to tour the campus. Fujisawa is a brand new campus of Keio University. In sharp contrast to other Japanese (and most US) universities, senior researchers at Keio teach freshmen and sophomores. Jun delights in telling how he will instruct a class of freshman, "now we will *ping* Switzerland."

At the Fujisawa campus, all entering freshmen are required to learn UNIX, to operate workstations, and acquire other basic skills. Although laptops are not required, they are very strongly encouraged.

**The library**

We walked into the library where students run an ID card through a reader to enter. Of course, Jun forgot his card, so we crawled over the barriers. We walked into a lounge area where groups of students sat clustered around HDTV sets watching assignments.

Other students were at workstations, each with a cassette tape drive and a headphone, doing language exercises. Other students were debugging programs. Finally, it dawned on me:

"Jun, there are no books in here."

"You still have books?" he asked with a smile. The books were up one floor, but it still gave the library a strange feeling.

Walking outside, we spotted Junsec patiently waiting for us in a car. We all went into Fujisawa where we were ushered into a private tatami room for an exquisite lunch of many courses, each beautifully arranged and delightfully prepared.

As I was gulping down the tenth course, a delicate broth with a single tiny mushroom and a little piece of fish, Jun started explaining to me how the spindly little mushroom I had just inhaled was handpicked in the forests and only available for a few weeks each year. Each mushroom cost as much as 5000 yen (US $35). I puffed out my cheeks as if I were still rolling the mushroom in my mouth to savor the delicate flavor, feeling like a Philistine.

**Akihabara**

Akihabara is a district of Tokyo devoted to consumer electronics, where building after building is stuffed with everything from 2x2 foot stalls to full-fledged, multifloor shops, each so full of devices that it would have made Marconi think he was hallucinating.

I knew I had reached Akihabara proper when I looked down on the sidewalk and saw a street vendor selling oscilloscopes and microchips. In a stall next to him, 60 different kinds of laptops were on sale, ranging from plain vanilla clones to full-fledged 80386, 60Mbyte, 6-pound notebooks, all for half the going US price.

Thinking I must have died and gone to geek heaven, I knew that this district was going to take a while to visit. I ducked into a noodle stand and used my two-word Japanese vocabulary to order a beer and a bowl of miso soup. As I noisily slurped my noodles, I could hear dozens of stereo systems, all playing different brands of disco at full volume.

Suitably fortified, I spent the next two hours trooping up and down stairs in building after building. I had to keep reminding myself that I had many more weeks on the road, lest I yield to the temptation to buy a home satellite dish (only $100) or a personal computer that weighed only 980 grams and ran on AA batteries. The computer was only available in a kanji model, but I figured I could always learn. Even more tempting were DAT drives, high definition TVs, global positioning system receivers, and three inch color televisions.

Feeling somehow unsatisfied, yet still solvent, I tore myself away and headed back towards the subway. On the way, I passed a group of young women all dressed up in natty uniforms bearing the somewhat cryptic label "With Me" and all walking in a line. The women at the head and tail of the column carried banners which matched their uniforms. The rest of the "With Me" girls all had the new two-pound Hitachi notebook, which they jauntily carried.

Every ten minutes or so, the conga line would go out into Akihabara and walk for a few blocks, dragging along a sound system gauranteed to attract lots of attention. They would wend their way back towards the "With Me" store and disappear inside, luring a portion of the mob in with them. Somehow, I suspected the short skirts had more to do with the crowd's loyalty than any desire to purchase Hitachi's latest and greatest.

**CARL MALAMUD** is currently living in Boulder, Colorado, a place he describes as "too close to Kansas and not far enough away from Marin County." He is the author of several technical reference books, and can be reached as carl@malamud.com.

# An Overview of Public-Key Cryptography Standards

by Burton S. Kaliski Jr., RSA Laboratories

**Abstract**

This article gives an overview of the PKCS family of standards for public-key cryptography. These standards cover RSA encryption, Diffie-Hellman key agreement, password-based encryption, extended-certificate syntax, cryptographic-enhancement syntax, and private-key information syntax. The article gives the motivation for the standards and discusses their relationship to other standards or agreements on public-key cryptography.

## Introduction

As public-key cryptography begins to see wide application and acceptance, one thing is increasingly clear: If it is going to be as effective as the underlying technology allows it to be, there must be interoperable standards. Even though vendors may agree on the basic public-key techniques, compatibility between implementations is by no means guaranteed. Interoperability requires strict adherence to an agreed-upon standard format for transferred data. The standards described here provide such a basis for interoperability.

We call the standards *Public-Key Cryptography Standards,* or "PKCS" for short. The standards consist of a number of components, called PKCS #1, #3, #5, #6, #7, #8 and #9. [Note 1]

The standards presented here evolved from the following broad design goals:

- To maintain compatibility with PEM (the *Internet Privacy-Enhanced Mail* (PEM) protocols [1–3]) wherever possible, at least to the extent of being able to share certificates and to translate encrypted and/or signed messages back and forth between PEM and PKCS.
- To extend beyond PEM in being able to handle arbitrary binary data (not just ASCII data), to handle a richer set of attributes in (extended) certificates, to handle Diffie-Hellman key agreement [4], and to handle a richer set of features in digitally signed and enveloped data.
- To propose a standard suitable for incorporation in future OSI (*Open Systems Interconnection* [5]) standards. The standards here are based on the use of OSI standard ASN.1 (*Abstract Syntax Notation One* [6]) and BER (*Basic Encoding Rules* [7]) to describe and represent data.

PKCS describes the syntax for messages in an abstract manner, and gives complete details about algorithms. However, it does not specify how messages are to be represented, though BER is the logical choice. Thus PKCS implementations are free to exchange messages in any manner, depending on character set, record size constraints, and the like, as long as the abstract meaning of the messages can be preserved from sender to recipient.

The PKCS standards are offered by RSA Laboratories to developers of computer systems employing public-key technology. It is RSA Laboratories' intention to improve and refine the standards in conjunction with computer system developers, with the goal of producing standards that most if not all developers adopt.

*Note: PKCS #2 and #4 are no longer active. PKCS #2 covered RSA encryption of message digests and PKCS #4 covered RSA key syntax. Both have been incorporated into the current PKCS #1.*

The role of RSA Laboratories in the standards-making process is five-fold:

- Publish carefully written documents describing the standards.
- Retain sole decision-making authority on what each standard is. This includes arbitrary object identifier choices, etc.
- Solicit opinions and advice from developers on useful or necessary changes and extensions.
- Publish revised standards when appropriate.
- Provide implementation guides and/or reference implementations.

Thus this standards-making process is not the usual committee-oriented method.

This article is divided into seven sections including this one. The next section gives some terminology. This is followed by a section which addresses the question, "What needs to be standardized?" After a summary of the PKCS family, we compare PKCS with other standards.

## Background information

This section gives the basic background information necessary to understand the terminology in this article. The information covers three areas: public-key cryptography, secret-key cryptography, and message-digest algorithms. For a more comprehensive background, the reader is referred to any of several nice survey articles [8, 9, 10].

## Public-Key Cryptography

*Public-key cryptography* is the technology first identified by Diffie and Hellman [4] in which encryption and decryption involve different *keys*. The two keys are the *public key* and the *private key*, and either can encrypt or decrypt data. A user gives his or her public key to other users, keeping the private key to himself or herself. Data encrypted with a public key can be decrypted only with the corresponding private key, and vice versa.

A *public-key algorithm* is an algorithm for encrypting or decrypting data with a public or private key. A private key is typically used to encrypt a message digest; in such an application, the public-key algorithm is called a *message-digest encryption algorithm*. A public key is typically used to encrypt a content-encryption key; in such an application, the public-key algorithm is called a *key-encryption algorithm*.

A *signature algorithm* is an algorithm that transforms a message of any length under a private key to a signature in such a way that it is computationally infeasible to find two messages with the same signature, to find a message with a given, predetermined signature, or to find the signature of a given message without knowledge of the private key. Typically, a signature algorithm is implemented by computing a message digest on the message, then encrypting the message digest with the private key.

*RSA* is a public-key algorithm invented by Rivest, Shamir, and Adleman [11] involving exponentiation modulo the product of two large prime numbers. The difficulty of breaking RSA is generally considered to be equal to the difficulty of factoring integers that are the product of two large prime numbers of approximately equal size.

*Key agreement* is a method whereby two parties, without prior arrangements, exchange messages in such a way that they agree upon a secret key that is known only to them. Key agreement can be achieved with a public-key algorithm, or with other methods. A *key-agreement algorithm* is an algorithm for achieving key agreement.

*continued on next page*

## Overview of Cryptography Standards *(continued)*

*Diffie-Hellman* is a key-agreement algorithm invented by Diffie and Hellman [4] involving exponentiation modulo a large prime number. The difficulty of breaking Diffie-Hellman is generally considered to be equal to the difficulty of computing discrete logarithms modulo a large prime number.

**Secret-key Cryptography**

*Secret-key cryptography* is the technology in which encryption and decryption involve the same key, a secret key. Pairs of users share a secret key, keeping the key to themselves. Data encrypted with a secret key can be decrypted only with the same secret key.

A *secret-key algorithm* is an algorithm for encrypting or decrypting data with a secret key. A secret key is typically used to encrypt the content of a message; in such an application, the key is called a *content-encryption key* and the secret-key algorithm is called a *content-encryption algorithm.*

A *password-based encryption algorithm* is a secret-key algorithm in which the key is derived from a user-supplied password.

The *Data Encryption Standard* (DES) is the standard federal secret-key algorithm [12].

**Message-digest algorithms**

A *message-digest algorithm* is a method of reducing a message of any length to a string of a fixed length, called the *message digest,* in such a way that it is computationally infeasible to find a collision (two messages with the same message digest) or to find a message with a given, predetermined message digest. *MD2* and *MD5* are message-digest algorithms invented by RSA/DSI [3, 13, 14]. Each inputs an arbitrary message and outputs a 128-bit message digest.

**What needs to be standardized?**

To answer the question, "What needs to be standardized?" we describe four applications of public-key cryptography: digital signature, digital enveloping, digital certification, and key agreement, looking at what aspects are suitable for standardization. Our emphasis is on those applications relevant to PKCS; there are certainly other applications, such as interactive authentication, that could be standardized.

The discussion assumes two independent levels of abstraction. The first level is message syntax, and the second level is specific algorithms. The intention is that message syntax and specific algorithms should be orthogonal. For example, a standard for the syntax of digitally signed messages should be able to work with any public-key algorithm, not just RSA; and a standard for RSA should be applicable to many different message syntax standards.

The description of the four applications involves the usual cryptographic players Alice and Bob.

**Digital signature**

*Digital signature* is an application in which a signer, say "Alice," "signs" a message $m$ in such a way that anyone can "verify" that the message was signed by no one other than Alice. The typical implementation of digital signature involves a message-digest algorithm and a public-key algorithm for encrypting the message digest (i.e., a message-digest encryption algorithm):

- Alice reduces the message $m$ to a message digest $d$ with a message-digest algorithm; then she encrypts the message digest $d$ with her private key, obtaining an encrypted message digest $\sigma$. She sends the message $m$ and the encrypted message digest $\sigma$ to Bob; the two parts together form the digitally signed message.

- Bob decrypts the encrypted message digest $\sigma$ with Alice's public key, obtaining the message digest *d;* then he reduces the message *m* to a comparative message digest *d'* and compares it to the message digest *d*. If the two are the same, he accepts the message.

Notice that Bob's work does not involve any information specific to him. Indeed, anyone can verify at any time that the message was signed by Alice, without access to any secret information. This application assumes that Bob knows Alice's public key; methods of developing trust in users' public keys are covered by the digital certificate application (see below).

Digital signature has three aspects that are suitable for standardization: an algorithm-independent syntax for digitally signed messages, specific message-digest algorithms, and specific public-key (message-digest encryption) algorithms.

Alice may also need a way to store her private key securely. One way to do this is to encrypt a message containing private-key information with a secret key derived from a password that Alice supplies. Aspects suitable for standardization here include an algorithm-independent syntax for encrypted private-key information, private-key syntax for specific public-key algorithms, and specific password-based encryption algorithms.

**Digital enveloping**

*Digital enveloping* is an application in which someone "seals" a message *m* in such a way that no one other than the intended recipient, say "Bob," can "open" the sealed message. The typical implementation of digital enveloping involves a secret-key algorithm for encrypting the message (i.e., a content-encryption algorithm) and a public-key algorithm for encrypting the secret key (i.e., a key-encryption algorithm):

- Alice encrypts the message *m* with a randomly generated secret key *k,* obtaining an encrypted message *c;* then she encrypts the secret key *k* with Bob's public key, obtaining an encrypted secret key $\varepsilon$. She sends the encrypted message *c* and the encrypted secret key $\varepsilon$ to Bob; the two parts together form the digitally enveloped message.
- Bob decrypts the encrypted secret key $\varepsilon$ with his private key, obtaining the secret key *k;* then he decrypts the encrypted message *c* with the secret key *k,* obtaining the message *m*.

Notice that Alice's work does not involve any information specific to her. Indeed, anyone can seal a message at any time for Bob, without access to any secret information. This application assumes that Alice knows Bob's public key; methods of developing trust in users' public keys are covered by the digital certificate application.

Digital enveloping has three aspects that are suitable for standardization: an algorithm-independent syntax for digitally enveloped messages, specific secret-key (content-encryption) algorithms, and specific public-key (key-encryption) algorithms.

Bob may need a way to store his private key securely, leading to similar aspects for standardization as those for digital signatures.

**Digital certification**

*Digital certification* is an application in which a certification authority "signs" a special message *m* containing the name of some user, say "Bob," and his public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in Bob's public key.

*continued on next page*

## Overview of Cryptography Standards *(continued)*

The typical implementation of digital certification involves a signature algorithm for signing the special message. (A signature algorithm is chosen here, rather than a message-digest algorithm followed by a message-digest encryption algorithm, as in the digital signature application, because X.509 certificates [15] only use a signature algorithm.)

- The certification authority transforms the special message $m$ to a signature $\sigma$ under a signature algorithm with the certification authority's private key. The certification authority sends the message $m$ and the signature $\sigma$ to some other user, say "Alice"; the two parts together form the digitally signed message, which is called a *certificate.*
- Alice verifies the signature $\sigma$ under the corresponding verification algorithm with the certification authority's public key. If the signature verifies, she accepts the message.

Notice that Alice's work does not involve any information specific to her. Indeed, anyone can verify at any time that the message was signed by the certification authority, without access to any secret information. Furthermore, anyone, not only the certification authority, can forward the message $m$ and the signature $\sigma$ to Alice.

This application assumes that Alice knows the certification authority's public key. Alice can develop trust in the certification authority's public key recursively, if she has a certificate containing the certification authority's public key signed by a superior certification authority whom she already trusts. In this sense, a certificate is a stepping stone in digital trust. Ultimately, one need only trust the public keys of a small number of top-level certification authorities. Through a chain of certificates, trust in a large number of users' signatures can then be established.

A broader application of digital certification includes not only Bob's name and public key but also other information about Bob in the special message $m$. Such a message, together with a signature, forms what PKCS terms an *extended certificate.* Extended certificates are more than stepping stones in digital trust. They enable the certification authority not only to give Alice a means of trusting Bob's public key, but also that other information. The other information may include, for example, Bob's electronic mail address, his authorization to sign documents of a given value, or his authorization to sign other certificates.

**CRL** A *certificate-revocation list* (CRL) is another type of special message together with a signature. The special message for a CRL contains a list of revoked certificates, where the certificates are typically referenced indirectly by a serial number. A CRL enables the certification authority to "void" its signatures on Bob's certificate or extended certificates, as might be required when Bob's name changes or his private key is compromised.

Digital certification has five aspects that are suitable for standardization: an algorithm-independent syntax for certificates, an algorithm-independent syntax for extended certificates, an algorithm-independent syntax for CRLs, public-key syntax for specific public-key algorithms, and specific signature algorithms.

### Key agreement

*Key-agreement* is an application in which Alice and Bob, without prior arrangements, exchange messages in such a way that they agree upon a secret key that is known only to them. The secret key can then be used, for example, to encrypt further communication between Alice and Bob. The typical implementation of key-agreement involves a two-phased key-agreement algorithm:

- Alice sends a message to Bob starting the key-agreement protocol.
- Alice and Bob independently perform a first phase of some key-agreement algorithm, and send the result of that phase to one another.
- Alice and Bob independently perform a second phase of the key-agreement algorithm, after which they arrive at a common agreed-upon secret key.

Key agreement has two aspects that are suitable for standardization: an algorithm-independent syntax for key-agreement messages, and specific key-agreement algorithms.

### Summary of useful standards

The foregoing discussion shows that the following standards are useful in implementing digital signature, digital enveloping, digital certification, and key agreement:

- *Algorithm-independent syntax:* digitally signed messages; digitally enveloped messages; certificates; extended certificates; certificate-revocation lists; encrypted private-key information; key-agreement messages.
- *Algorithm-specific syntax:* public keys; private keys.
- *Algorithms:* message digest; secret-key encryption; public-key encryption; signature; password-based encryption; key agreement.

## The PKCS standards

This section describes the members of the PKCS family. The descriptions are largely taken from the PKCS documents themselves. PKCS leaves ample room for future expansion. Most objects defined by PKCS carry version numbers to allow backward compatibility in future revisions. Several of the objects also have space for arbitrary "attributes" that carry additional information not directly addressed by PKCS.

### PKCS #1: RSA Encryption

PKCS #1 [17] describes a method, called *rsaEncryption,* for encrypting data using the RSA public-key cryptosystem [11]. Its intended use is in the construction of digital signatures and digital envelopes, as described in PKCS #7:

- For digital signatures, the content to be signed is first reduced to a message digest with a message-digest algorithm (such as MD5 [14]), and then an octet string containing the message digest is encrypted with the RSA private key of the signer of the content. The content and the encrypted message digest are represented together according to the syntax in PKCS #7 to yield a digital signature. This application is compatible with Privacy-Enhanced Mail methods [1,3].
- For digital envelopes, the content to be enveloped is first encrypted under a content-encryption key with a content-encryption algorithm (such as DES [12]), and then the content-encryption key is encrypted with the RSA public key(s) of the recipient(s) of the content. The encrypted content and the encrypted content-encryption key are represented together according to the syntax in PKCS #7 to yield a digital envelope. This application is compatible with Privacy-Enhanced Mail methods.

*continued on next page*

## Overview of Cryptography Standards *(continued)*

PKCS #1 also describes a syntax for RSA public keys and private keys. The public-key syntax would be used in certificates; the private-key syntax would be used typically in encrypted private keys (PKCS #8). The public-key syntax is identical to that in both X.509 [15] and PEM [3]. Thus X.509/PEM RSA keys can be used in PKCS #1.

PKCS #1 further defines two signature algorithms, called *md2with - RSAEncryption* and *md5withRSAEncryption,* for use in signing X.509/ PEM certificates and certificate-revocation lists [2, 15], PKCS #6 extended certificates, and other objects employing digital signatures such as X.400 message tokens [28].

**PKCS #3: Diffie-Hellman Key Agreement**

PKCS #3 [18] describes a method for implementing Diffie-Hellman key agreement [4], whereby two parties, without any prior arrangements, can agree upon a secret key that is known only to them (and, in particular, is not known to an eavesdropper listening to the dialogue by which the parties agree on the key). This secret key can then be used, for example, to encrypt further communications between the parties. The intended application of PKCS #3 is in protocols for establishing secure connections, such as those proposed for the transport layer and the network layer.

**PKCS #5: Password-based Encryption**

PKCS #5 [19] describes a method for encrypting an octet string with a secret key derived from a password. The result of the method is an octet string. Although PKCS #5 can be used to encrypt arbitrary octet strings, its intended primary application to public-key cryptography is for encrypting private keys when transferring them from one computer system to another, as described in PKCS #8.

PKCS #5 defines two key-encryption algorithms: *pbeWithMD2 - AndDES-CBC* and *pbeWithMD5AndDES-CBC.* The algorithms employ DES secret-key encryption [12] in cipher-block chaining mode [20], where the secret key is derived from a password with the MD2 or MD5 message-digest algorithm [3,13,14].

**PKCS #6: Extended-Certificate Syntax**

PKCS #6 [21] describes a syntax for extended certificates. An extended certificate consists of an X.509 public-key certificate [15] and a set of attributes, collectively signed by the issuer of the X.509 public-key certificate. Thus the attributes and the enclosed X.509 public-key certificate can be verified with a single public-key operation, and an ordinary X.509 certificate can be extracted if needed, e.g., for Privacy-Enhanced Mail [1, 2].

The intention of including a set of attributes is to extend the certification process beyond just the public key to include other information about a given entity, such as electronic-mail address. A non-exhaustive list of attributes is given in PKCS #9. The preliminary intended application of PKCS #6 is in the cryptographic-enhancement syntax standard (PKCS #7), but it is expected that other applications will be developed.

**PKCS #7: Cryptographic Message Syntax**

PKCS #7 [22] describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes. The syntax admits recursion, so that, for example, one envelope can be nested inside another, or one party can sign some previously enveloped digital data. It also allows arbitrary attributes, such as signing time, to be authenticated along with the content of a message, and provides for other attributes such as countersignatures to be associated with a signature. A degenerate case of the syntax provides a means for disseminating certificates and certificate-revocation lists.

PKCS #7 is compatible with PEM [1] in that signed-data and signed-and-enveloped-data content, constructed in a PEM-compatible mode, can be converted into PEM messages without any cryptographic operations. PEM messages can similarly be converted into the signed-data and signed-and-enveloped data content types.

PKCS #7 can support a variety of architectures for certificate-based key management, such as the one proposed for Privacy-Enhanced Mail in RFC 1114 [2]. Architectural decisions such as what certificate issuers are considered "top-level," what entities certificate issuers are authorized to certify, what distinguished names are considered acceptable, and what policies certificate issuers must follow (such as signing with secure hardware, or requiring entities to present specific forms of identification) are left outside PKCS #7.

The values produced according to PKCS #7 are intended to be BER-encoded [7], which means that the values would typically be represented as octet strings. While many systems are capable of transmitting arbitrary octet strings reliably, it is well known that many electronic-mail systems are not. PKCS #7 does not address mechanisms for encoding octet strings as (say) strings of ASCII characters or other techniques for enabling reliable transmission by re-encoding the octet strings. RFC 1113 [1] suggests one possible solution to this problem.

**PKCS #8: Private-Key Information Syntax**

PKCS #8 [23] describes a syntax for private-key information. Private-key information includes a private key for some public-key algorithm and a set of attributes. PKCS #8 also describes a syntax for encrypted private keys. A password-based encryption algorithm (e.g., one of those described in PKCS #5) could be used to encrypt the private-key information.

The intention of including a set of attributes is to provide a simple way for a user to establish trust in information such as a distinguished name or a top-level certification authority's public key. While such trust could also be established with a digital signature, encryption with a secret key known only to the user is just as effective and possibly easier to implement. A non-exhaustive list of attributes is given in PKCS #9.

**PKCS #9: Selected Attribute Types**

PKCS #9 [24] defines selected attribute types for use in extended certificates (PKCS #6), digitally signed messages (PKCS #7), and private-key information (PKCS #8).

**Compatibility with other work**

This section describes the compatibility of the PKCS standards with other standards or agreements on public-key cryptography. For simplicity, we refer to the various works involving public-key cryptography as "standards," without regard to their formal approval by a standards-making body. Compatibility has many meanings. For instance, a standard *A* may be considered compatible with another standard *B* if standard *A* provides algorithms that standard *B* can use. Or, standard *A* may generate data that standard *B* can process directly. We choose the definition that standard *A* is compatible with standard *B* if standard *A* provides something useful to standard *B*, where the usefulness may be contingent on a change in representation, and possibly on omission of information. Cryptographic operations are not allowed in the change of representation.

We say standard *A* is "outbound" compatible with standard *B* if implementations of standard produce something useful to implementations of standard *B*, but not necessarily vice versa.

*continued on next page*

## Overview of Cryptography Standards *(continued)*

We say standard *A* is "inbound" compatible with standard *B* if implementations of standard *B* produce something useful to implementations of standard *A*, but not necessarily vice versa.

We address compatibility with two standards:

- Privacy-Enhanced Mail, as defined in RFCs 1113–1115 [1–3] and their successors [25, 16, 26].
- Directory Services—Authentication Framework, as defined in CCITT Recommendation X.509 [15].

### Privacy-Enhanced Mail

PKCS is inbound compatible with Privacy-Enhanced Mail, as defined in the proposed successors to RFCs 1113–1115 [25, 16, 26]. With suitable restrictions, PKCS is outbound compatible as well. However, PKCS, like the proposed successors to RFCs 1113–1115, is incompatible with the original RFCs 1113–1115 [1–3]. (The successors to RFC 1113–1115 are currently Internet Draft documents. Unlike RFCs these have no formal standing. It is expected, though not guaranteed, that the next set of PEM RFCs will be essentially the same as the current Internet Drafts.)

A privacy-enhanced message generated according to the proposed successors to the Privacy-Enhanced Mail RFCs can be converted to a form that can be processed by implementations of PKCS #7 without any cryptographic operations. The conversion process is "flat" in the sense that the encapsulated text of the privacy-enhanced message becomes the "inner" content of the PKCS #7 data. If the encapsulated text happens to contain privacy-enhanced messages, those messages are not interpreted in the conversion process.

Data with certain PKCS #7 cryptographic enhancements can be converted to a form that can be processed by implementations of the proposed successors to the Privacy-Enhanced Mail RFCs.

Privacy-Enhanced Mail can effectively be viewed as a set of encoding rules, analogous to the Basic Encoding Rules for ASN.1, for PKCS #7 data with these restrictions. Conversion from PKCS #7 to PEM may involve omission of attributes from PKCS #6 extended certificates, which is acceptable since the attributes are not essential to PEM.

RSA encryption in PKCS #1, in block types 01 and 02, is the same as in PEM, as defined by the latest Internet-Draft successor to RFC 1115 [26]. Certificates in PEM are one of the alternatives of PKCS #7's *ExtendedCertificateOrCertificate* type. (See the next section for more details.) The *md2WithRSAEncryption* and *md5WithRSAEncryption* signature algorithms in PKCS #1 are the same as PEM's message and certificate signature algorithms (also defined by the Internet Draft).

### Directory Services Authentication Framework

PKCS is compatible with Directory Services—Authentication Framework, as defined in CCITT Recommendation X.509 [15]. A certificate generated according to X.509 (as revised by RFC 1114 [2]) can be converted to a form that can be used in implementations of PKCS #7. The conversion involves the type *ExtendedCertificateOrCertificate,* which has two alternatives, an X.509 certificate and a PKCS #6 extended certificate. An extended certificate generated according to PKCS #6 can be converted to a form that can be used in implementations of X.509 (as revised by RFC 1114). The conversion involves the omission of extended attributes.

RSA private-key encryption in PKCS #1 is the same, in block type 00, as RSA private-key encryption in X.509.

The signature process for X.509 certificates is the same as the signature process for PKCS #6 extended certificates. That is, both use X.509's SIGNED macro (or an equivalent form), so both can use any signature algorithm consistent with the SIGNED macro.

The *md2WithRSAEncryption* and *md5WithRSAEncryption* signature algorithms in PKCS #1 are consistent with the SIGNED macro, in that they input an octet string and output a bit string. Thus, they can be used in signing X.509 certificates, or any other quantity signed in the authentication framework or in other uses of the SIGNED macro (e.g., in X.411 security [28].

RSA public-key syntax in X.509 Annex C is the same as RSA public-key syntax in PKCS #1.

**Security conditions**

The three algorithm standards—PKCS #1 (RSA Encryption Standard), PKCS #3 (Diffie-Hellman Key Agreement standard), and PKCS #5 (Password-Based Encryption Standard)—all involve security conditions on the choice of key (or password, in the case of PKCS #5). Such conditions may change as the state of the art in cryptanalysis improves, and are subject to tradeoffs between performance and security. For example, the conventional argument that the factors of the RSA modulus should be strong primes seems no longer to hold [29], which is why PKCS neither mandates strong primes, nor discourages their use.

Since security conditions do not affect the format of transferred data, the security conditions are left outside the scope of PKCS. Specific open issues, left to implementors, include:

- Range of lengths of RSA modulus $n$ in PKCS #1 (for example, RFC 1114 sets the range as 320 to 632 bits [2], and the proposed successors to RFC 1115 set the range as 508 to 1024 bits [26])
- Conditions on RSA primes $p$ and $q$, such as whether $p-1$ and $q-1$ should have large factors, and how far apart $p$ and $q$ should be
- Additional conditions on the RSA public exponent $e$ and the RSA private exponent $d$
- Range of lengths of the Diffie-Hellman modulus $p$ in PKCS #3
- Conditions on the Diffie-Hellman modulus $p$, such as whether $p-1$ should have a large factor
- Conditions on the Diffie-Hellman base $g$, such as how large a group it should generate (e.g., all non-zero elements modulo $p$)
- Range of lengths of the password $P$ in PKCS #5
- Structural requirements on the password $P$ (e.g., at least one non-alphanumeric character)
- Sources of pseudorandom bits in all the algorithm standards

It is RSA Laboratories' intention to release "recommended practices" documents from time to time that address security conditions such as those just listed.

**Summary**

The PKCS family of standards addresses the following need: an agreed-upon standard format for transferred data based on public-key cryptography.

*continued on next page*

## Overview of Cryptography Standards *(continued)*

PKCS covers several aspects of public-key cryptography, including RSA encryption, Diffie-Hellman key agreement, password-based encryption, extended-certificate syntax, cryptographic-enhancement syntax, and private-key information syntax. PKCS evolved from three broad design goals: to maintain compatibility with Privacy-Enhanced Mail, to extend beyond PEM, and to be suitable for incorporation in future OSI standards.

### Conclusion

This article has summarized PKCS. It has shown that PKCS provides a basis for interoperability in the several areas of interest, and that PKCS has a high level of PEM compatibility, several extensions, and significant compatibility with existing OSI standards. The reader is encouraged to review and implement PKCS and to make constructive comments.

### References


[1] J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures," RFC 1113, August 1989. See also [28].

[2] S. Kent & J. Linn. "Privacy Enhancement for Internet electronic mail: Part II—Certificate-based key management," RFC 1114, August 1989. See also [16].

[3] J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, Modes, and Identifiers," RFC 1115, August 1989. See also [29].

[4] W. Diffie & M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,* IT-22:644–654, 1976.

[5] CCITT, "Recommendation X.200: Reference Model of Open Systems Interconnection for CCITT Applications," 1984.

[6] CCITT, "Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1)," 1988.

[7] CCITT, "Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)," 1988.

[8] Ronald L. Rivest, "Cryptography," In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science,* Volume 1, pages 719–755, Elsevier Science, 1990.

[9] W. Diffie, "The first ten years of public-key cryptography," *Proceedings of the IEEE,* 76(5):560–577, May 1988.

[10] W. Diffie & M.E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings of the IEEE,* 67(3):397–427, March 1979.

[11] R. L. Rivest, A. Shamir, & L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM,* 21(2):120–126, February 1978.

[12] National Bureau of Standards, "Data Encryption Standard," (FIPS Publication 46–1), January 1988.

[13] B. S. Kaliski Jr., "The MD2 Message-Digest Algorithm," I-D: `draft-rsadsi-kaliski-md2-01.txt`

[14] Ronald L. Rivest, "The MD5 Message-Digest Algorithm," I-D: `draft-rsadsi-rivest-md5-02.txt`

[15] CCITT, "Recommendation X.509: The Directory—Authentication Framework," 1988.

[16] S. Kent, "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management,"
I-D: `draft-ietf-pem-keymgmt-00.txt`

[17] RSA Data Security, Inc., "PKCS #1: RSA Encryption Standard," Version 1.4, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-18, June 1991.)

[18] RSA Data Security, Inc., "PKCS #3: Diffie-Hellman Key-Agreement Standard," Version 1.3, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-19, June 1991.)

[19] RSA Data Security, Inc., "PKCS #5: Password-Based Encryption Standard," Version 1.4, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-20, June 1991.)

[20] National Bureau of Standards, FIPS Publication 81: "DES Modes of Operation," December 1980.

[21] RSA Data Security, Inc., "PKCS #6: Extended-Certificate Syntax Standard," Version 1.4, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-21, June 1991.)

[22] RSA Data Security, Inc., "PKCS #7: Cryptographic Message Syntax Standard," Version 1.5, February 1992. (Earlier version published as NIST/OSI Implementors' Workshop document SEC-SIG-91-22, June 1991.)

[23] RSA Data Security, Inc., "PKCS #8: Private-Key Information Syntax Standard," Version 1.1, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-23, June 1991.)

[24] RSA Data Security, Inc., "PKCS #9: Selected Attribute Types," Version 1.0, June 1991. (Also published as NIST/OSI Implementors' Workshop document SEC-SIG-91-24, June 1991.)

[25] J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures," I-D: `draft-ietf-pem-msgproc-01.txt`.

[26] D. Balenson, "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers,"
I-D: `draft-ietf-pem-algorithms-01.txt`

[27] CCITT, "Recommendation X.400: Message Handling System and Service Overview," 1988.

[28] CCITT, "Recommendation X.411: Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures," 1988.

[29] Rivest, Ronald, L., "Are 'strong' primes needed for RSA?" Unpublished manuscript, April 1991.

[30] *ConneXions,* Volume 4, No. 8, August 1990, "Special Issue on Network Management and Network Security."

***Learn more about this topic in tutorial T25: "Theory of Cryptography and its application to Network Security," (Monday–Tuesday) and in session S1: "Competing Encryption Technologies" Wednesday, May 20 at 10:30am.***

***Author's address:***

Burton S. Kaliski Jr.
RSA Data Security, Inc.
10 Twin Dolphin Drive
Redwood City, CA 94065
USA
Tel.: 415-595-8782
Fax: 415-595-1873

**BURTON S. KALISKI** Jr. received the B.S., M.S., and Ph.D. degrees in computer science from MIT. In 1989, he joined RSA Data Security, where he is currently chief scientist of the RSA Laboratories division. His research interests include cryptography and fast arithmetic techniques. He was a Visiting Assistant Professor at Rochester (New York) Institute of Technology during 1988–89. Dr. Kaliski is a member of the IEEE Computer Society, the International Association for Cryptologic Research, Sigma Xi, and Tau Beta Pi. He was general chair of *CRYPTO '91.* He can be reached as `burt@rsa.com`.

# TCP/IP—The Hero of Operation Desert Storm Information Systems

by Robert F. Weissert, OAO Corporation

Just as the Patriot missile became the hero of the Army's Desert Storm weapons systems, the *Transmission Control Protocol/Internet Protocol* (TCP/IP) became the hero of Operation Desert Storm's Information Systems.

## Planning

Within six weeks of the Iraqi invasion of Kuwait, the *U.S. Army Information Systems Command* (USAISC) had designed, installed and made fully operational a data communications architecture in Saudi Arabia using off-the-shelf commercial products. There were no MILNET or NSFNET access points in Saudi Arabia so the first problem was how to access the Internet backbone. We basically had 2 choices, we could request satellite links back to a MILNET *Packet Switching Node* (PSN) in the United States or try to obtain a link to the DDN in Europe. At the time, European military units were not involved, so the decision was made to request 56Kbps links to the MILNET in the United States. *Defense Information Systems Agency* (DISA) engineers had to "model" this architecture to ensure that the selected PSNs had the capacity and throughput to support the Desert Storm traffic. This was very difficult to model since no one knew how much or what type of traffic was involved. At this point, we had to estimate what services and agencies would be using this network and what types of traffic they would be sending.

The staff of the Information Systems Operations Branch at Ft. Huachuca, AZ, spent countless nights and weekends coordinating with their Air Force, Navy, and Marine Corps counterparts to determine requirements. Mr. Tony Fogle, the current Army Internet Manager wrote a program to download and analyze IP destination addresses to establish a baseline. With all of this information in hand, some very smart engineers in DISA were able to model the network.

While waiting for the satellite links, we obtained permission from Central Command (CENTCOM) information officer to use a dedicated voice line which was already installed from Ft. Bragg, NC to Dhahran, Saudi Arabia. I remember distinctly when Captain Bob Davenport from the Army Information Systems Engineering Command sat in Dhahran and connected a 9.6Kbps modem to this line, converting it to an X.25 MILNET access link. For the first 6 weeks, every IP datagram destined for Saudi Arabia traveled over this 9.6Kbps line. Twice in one week, the line was severed by tanks and all data stopped until it was repaired. Until the "batmobiles" arrived, this 9.6Kbps line would have to handle all of the traffic flowing from Saudi Arabia to the rest of the Internet. Batmobiles are mobile satellite ground terminals produced by a commercial vendor. It's basically a Winnebago with a dish antenna on the roof. John Sullivan, Dick Hagen, Maj. Noel Goyette, and countless others spent weeks trying to get airlift for those vans.

## Class B network addresses

The next step was to request 20 class B network addresses from the NIC. The NIC issued these addresses over the phone in about 15 seconds. Cooperation from the NIC was paramount to our success and they never hesitated giving it. We now had 20 unique IP network addresses and one 9.6Kbps line to Saudi Arabia. The next (and biggest) challenge was getting equipment on military flights to Saudi Arabia. We needed to send Cisco AGS/2 routers, AT&T 3B2 minicomputers, PCs, modems, etc. to Dhahran to set up the first node (ODS-HOST1).

We couldn't get space on military transport planes since they were busy moving tanks and troops to reinforce the small and vulnerable ground force already in place. Commercial air travel seemed to be the best answer. We bought tickets on commercial flights to Riyadh and Dhahran, and sent the people who were ultimately responsible for the success of our plan—"the USAISEC engineers"—to Saudi Arabia carrying the equipment as "excess baggage." We also used Federal Express to send Next Day Air packages to the area.

Within 10 days, the mailbridge at BBN was receiving its first EGP updates from Dhahran, Saudi Arabia, and the Army had established "E-Mail To The Foxhole," a term coined by the Army in Europe.

**Deployment**

For the next 8 weeks (October–November 1991), Army engineers in Saudi Arabia worked hand-in-hand with engineers and technicians from all services, from DISA and from academia to install an "internet" environment consisting of 20 class B subnetworks at key locations in Saudi Arabia. Riyadh, the nation's capital and the headquarters of the U.S. Central Command, Dhahran, and King Khalid Military City (KKMC) were the first 3 nodes. The U.S. Army VII Corps deployed from Germany to their staging area in KKMC.

This network proved very useful for connecting heterogeneous systems together, but it was by no means the solution to all the data exchange problems. Soldiers and airman still carried floppy disks from PCs to an "Internet Host." Many computers didn't and couldn't run TCP/IP. There were hundreds of loose ends. It did work however, and its success was predicated by the strategic commitment 6 years ago to employ reliable Internet protocols worldwide. At that time, Lieutenant General Emmett Paige Jr., the Commanding General of USAISC approved the purchase and installation of the entire TCP/IP protocol suite on major Army processing centers which previously used only IBM's SNA and Bisync architecture. These systems included the Army finance center, the Army logistics centers and the Army's Regional Data Centers, which provided the data processing for most major Army installations in the continental United States. All in all, the Army installed TCP/IP on over 1500 mini and mainframe computers worldwide.

**Applications**

The Army's TCP/IP network in Saudi Arabia provided worldwide connectivity for all types of computers and operating systems. It was robust and dynamic enough to satisfy battlefield requirements for rapid change and it was reconfigured in "real-time" as the situation changed. Routing, protocol, security and addressing changes were made by CW3 Art Olson in Saudi Arabia and by engineers in the U.S. and Germany. In addition to the standard services (FTP, SMTP, Telnet) we ran IBM's 3270 and 2780 (RJE) protocol over the TCP/IP channels. EDS engineers spent Christmas of '91 in Dhahran trying to pass 2780 RJE data over TCP/IP.

Users of the Army's Desert Storm network experienced throughput in excess of over 50 megabytes a day, and interactive response time back to the United States of less than three seconds. This network was used by Army, Navy, Air Force and Marine elements in Saudi Arabia for unclassified logistics data. It was also used by troops to send and receive notes to their wives and families around the globe. It was a significant complement to the existing "Command and Control" systems which actually controlled the battle.

*continued on next page*

## TCP/IP—The Hero of Desert Storm *(continued)*

**Problems**

Besides tanks running over cables, we encountered two technical problems. The first problem involved EGP tables in all of the routers. They suddenly became corrupted. The only network that we could reach was 26 (MILNET) which wasn't difficult since that was the network we were directly connected to and listed as our "route of last resort." We were communicating with EGP core gateways but the downloaded tables were corrupted. Mr. Leo Scherping, located at Ft. Ritchie, MD, worked on this problem around the clock. We managed to fix the problem temporarily by using static routing tables and TFTP to transfer the "last known good" table to the routers. DISA (DCA) corrected the mailbridge problem in about 48 hours.

The other problem was also EGP related but was a basic design flaw. To "split the load" and provide redundancy, we connected each router to 2 different PSNs on network 26. Our routers didn't understand 2 access points to the same network, they broadcasted reachability information based on the first MILNET address in their configuration files. In addition, the NIC static host tables (HOSTS.TXT) only listed one Class A MILNET address for the Desert Storm Class B subnetted networks. The *Border Gateway Protocol* (BGP) or a more dynamic OSI external routing protocol should prevent this problem in the future, but Mr. Luis Morales, an Army engineer, gave us a work-around. As the figure shows the network quickly started to get complicated. Every unclassified computer in Saudi Arabia wanted access to the Internet. We even had IBM 3278 terminal users asking if they could get an SDLC link to their 3090 Front End Processor in the Pentagon.

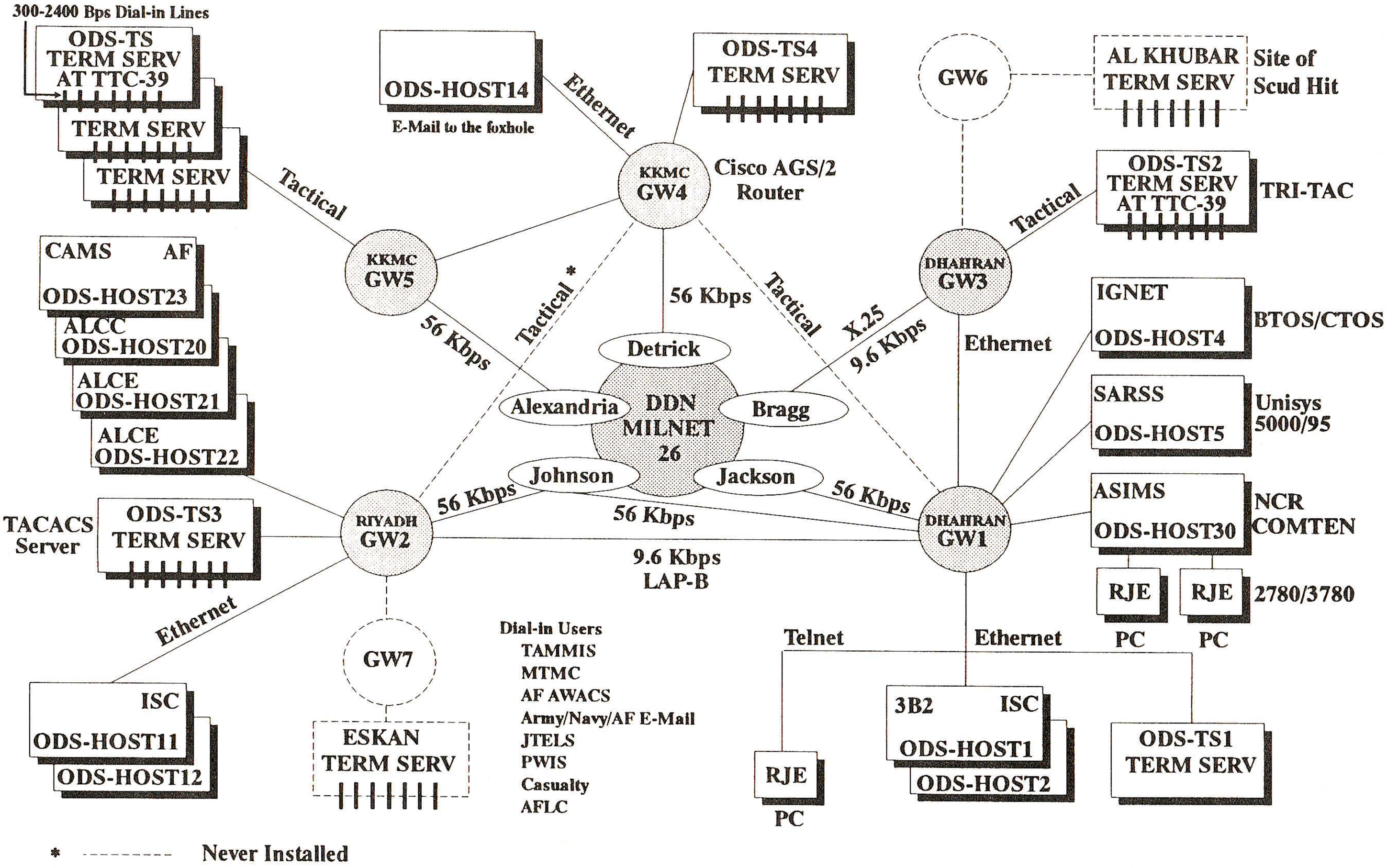
This complex collection of routers, bridges, packet switches and satellite trunks required tedious, constant and accurate network management from a centralized location. If the Defense Information Systems Agency and the Department of Defense learned anything about networking during Desert Storm, it was that global computer networks must be managed by a central authority. "Rapid Prototyping" for strategic applications using fourth generation languages (4GL), artificial intelligence and computer aided software engineering tools were used to quickly develop an application called the "Global Internet Information System." This UNIX-based application was prototyped at the Cambridge Technology Group and was developed into a fully functional application by Maj. John Buono, Ph.D. This application was used to monitor 2000 Army computers on 65 heterogeneous networks around the world, including the "Operation Desert Storm" network.

**Security**

Security was a serious concern. With memories of the "Morris Worm" and the "Cuckoo's Egg" still fresh in our minds, the designers and implementors of the Desert Storm network were very cautious about security. Anonymous logins were strictly forbidden. TFTP was used only when necessary and FTP required 2 passwords. The terms "Desert Shield," "Saudi Arabia," "South West Asia" or "Kuwait" were never used to register hosts or users. The only clue we gave to anyone with access to WHOIS or HOST.TXT was the term "ODS." We named all the hosts, gateways, TACS, etc. ODS-HOST1 or something similar. Although there have been reports to the contrary, I never encountered an unauthorized access of data during the 8 months the network served the coalition forces. Since there were no BBN C/30 Internet TACS in Saudi Arabia, TAC USERIDS were basically useless.

# Operation Desert Storm

## Data Communications Architecture




*continued on next page*

## TCP/IP—The Hero of Desert Storm *(continued)*

The Army had installed its own terminal (Client Telnet) servers in 3 or 4 locations with dial in lines through both commercial circuits and tactical circuits. These terminal servers supported the TACACS client/server protocol but they didn't operate with the installed base of TAC USERIDS issued by the NIC. To solve this problem, we downloaded C source code from Cisco's BBS in Menlo Park, CA., and sent it to a programmer (Bruce Funk) in Germany. Bruce compiled it to run on an AT&T 3B2 UNIX computer and Frank Wancho of White Sands Missile Range set up a TACACS server at Ft. Huachuca, AZ, and backups at the U.S. Military Academy, West Point and White Sands. The Army Domain Name Server which responded to port 42 DNS calls was also installed at West Point and was operated by LTC Steve Lowe and Mr. Patrick McGuinness. Most of the Army at this time was still using static name resolution by downloading HOSTS.TXT twice a week. The use of DNS instead of static host tables was first authorized for general Army use during Desert Shield.

### Conclusion

In 1974, the *Defense Advanced Research Projects Agency* (DARPA), together with USC, UC Berkeley and many others began building the ARPANET, defining and standardizing TCP/IP and writing RFCs. By 1985, a fully functional Internet existed and was being used by academia and the military. Between 1986–1990, the U.S. Army installed TCP/IP protocols on IBM mainframes at its finance centers, supply centers, and central data processing centers. By December 1991 the terms "Open Systems" and "Interoperability" were no longer buzz words in the Defense Department. 15 years of research and testing by DARPA, academia, the military services and commercial firms translated into a fully functional computer network in Saudi Arabia.

**ROBERT F. WEISSERT** is a Senior System's Engineer for OAO Corporation headquartered in Greenbelt MD. Mr. Weissert recently retired from the U.S. Army where he served as the Army Internet manager for the Army's Information Systems Command (USAISC) at Ft. Huachuca, AZ. While assigned to USAISC, Mr. Weissert managed the "Operation Desert Storm" computer network in Saudi Arabia. This network integrated and provided interoperability for over 20 IBM MVS mainframe systems and 800 UNIX based minicomputer systems. He also served as the technical advisor to the Army's Electronic Mail Configuration Control Board. Prior to his assignment to the Army Information Systems Command, Mr. Weissert was assigned to the White House Communications Agency in Washington, D.C., where he served three Presidents. Mr. Weissert has lectured on systems integration and open systems architecture at the U.S. Military Academy, West Point, at the U.S. Army Computer Science Schools in the U.S. and Europe, and at IEEE and AFCEA conferences worldwide. His military awards include the Legion of Merit and the Presidential Service Badge. Mr. Weissert received his degree from the University of New York. He can be reached as `bob@huachuca-giis.army.mil`.

***You can learn much more about the use of TCP/IP in Operation Desert Storm during INTEROP 92 Spring. Bob Weissert and Tony Fogle will host a Birds of a Feather session on this topic on Thursday night, May 21 at 7:30pm. Also, don't miss General H. Norman Schwarzkopf's plenary address "Desert Storm: Lessons Learned for All of Us" on Friday morning at 8:30am.***

# Profile: RAINet: The Research And Information Network

by Steve Neighorn, Randy Bush, and Jeff Beadles

## Background and introduction

The Portland, Oregon, metropolitan area has one of the largest per-capita concentrations of private UNIX systems in the United States. These systems are usually operated by an individual and support small user communities. The systems range from Intel 80x86 systems running System V flavors of UNIX to Macs running A/UX to Tektronix and Sun Microsystems workstations. In the past, connectivity between these systems, for the transfer of electronic mail and news, was exclusively based on *uucp*.

Some number of years ago, several of the local private-site system administrators wanted to form a domain to ease e-mail addressing problems and get away from the `.UUCP` zone. But the lack of local sites willing to do MX-forwarding, along with the costs of long-distance phone calls and IP connectivity, and the usual social issues involved with trying to get a diverse group to agree on a single plan, doomed those early attempts.

In April of 1990, several local admins received notice of AlterNet's "IP Networking for the rest of us": a project that put network connections into various cities (including Portland) with a cost factor which would allow small and medium sized organizations the opportunity to take advantage of both AlterNet's own network, and if desired, the NSF Internet.

The local sysadmins once again began working on ideas to get their own machines into a registered domain, and if possible, connect the various sites together and hook up to other larger networks.

Several pizza-based meetings were held, and a plan developed. Funding was a primary concern, because the initial charge for hooking up a 56Kbps line to AlterNet was close to $10000, with monthly recurring charges hovering around $1500. Finally a deal was struck with UNINET-ZA (an Academic and Research Network in Southern Africa) in which we received funding in return for engineering work on some of the low-cost IP connectivity technologies, and a connection for them.

RAINet first went online in January of 1991. These connections were local only. RAINet then purchased the 56Kbps service from AlterNet, and in June of 1991 connected to the outside world. AlterNet placed a restriction on the number of sites whose packets would be allowed to pass through to AlterNet. Four sites in RAINet are directly on the Internet, and they act as a firewall or gateway to the rest of RAINet. Sites behind the firewalls cannot get to the Internet, and Internet sites can only reach those machines at the four firewall sites.

## Hierarchical structure

RAINet can be thought of as a hierarchical structure, with the AlterNet connection at the top, connecting to the Cisco router, PSGnet (a private network) and the RAINet routers at the next level, which in turn connect to five other sites (three of which are on the Internet), which in turn connect to between zero and three other sites, and so on.

RAINet maintains several internal mailing lists, including lists for everyone in the `.rain.com` domain, those who are IP connected, those who founded and continue to gently "guide" RAIN, and a security list.

*continued on next page*

## Profile: RAINet *(continued)*

Once every two to three months, all interested parties gather for an informal tech/policy/dinner discussion. So far, there have been few personality problems, and we have all managed to stay friendly with one another.

**Goals**

RAINet was established to experiment with medium distance network technologies, and to improve electronic communications among local Portland-metro area UNIX systems and users.

Although RAINet is small, it consists of a full range of users, including engineers, researchers, educators, consultants, and hobbyists.

RAINet users have examined several "thin wire" interconnect topologies, including standard phone lines, inexpensive leased lines, packet radio, and cable TV.

In addition, RAINet experiments with low-cost hardware options for routers, such as cheap PC-based equipment, host-based IP protocols like SLIP and PPP, and medium speed modems (V.32, PEP, and other proprietary systems).

RAINet is explicitly not designed to provide production level services in the fashion of AlterNet, CompuServe, or TymNet. As an experimental network using common phone lines, inexpensive equipment, and run by volunteers with daytime jobs, there will frequently be times when parts of the network are inoperative.

RAINet will provide non-commercial connectivity to the Internet research and educational electronic communities, as a number of RAINet's founders are involved in the research with educational and research institutions outside of the Pacific Northwest.

**Guidelines**

RAINet has drafted and uses an acceptable-use statement similar to that of the CREN. It is modified in that RAINet does not attempt to provide production level service, and light commercial traffic will be allowed within (but not outside of) RAINet.

RAINet member sites and users are required to take reasonable measures, given the constraints of technology and management, to ensure that traffic using gateways between RAINet and other networks conforms to the guidelines of the other networks.

**Status**

RAINet went online with its first connection in early January 1991. Currently there are more than 16 sites connected via SLIP (Ethernet is used within each site). These IP connected sites have over 70 hosts including work stations, X-terminals, PCs, and routers. In addition, RAINet provides MX-forwarding to dozens of other local sites within the rain.com domain in the Portland-metro area.

**Network makeup**

The RAINet-to-AlterNet link is a 56Kbps line connected via DSUs to a Cisco router. The router sits on an Ethernet with three other 80286-based PC-routers running KA9Q. Each PC-router has an Ethernet card and two serial ports utilizing 16550AFN UARTs. The UNINET link is accomplished with a 14.4Kbps modem which is plain old V.32bis just like many of the others. The other connections are accomplished using either V.32 or V.32bis modems, including Intel and Telebit products. Each of the NSF-connected and several of the downstream sites connected to the NSF-connected (firewall) sites also use PC-routers. The standard hardware is a faster (12MHz+) 80286 or 80386SX PC clone, monochrome monitor and mono card, keyboard, high density floppy disk, WD8003 Ethernet card, 640k of DRAM, and one to four serial ports using the faster 16550 UART.

Software on the PC-routers is either a customized version of KA9Q or a customized version of PC/ROUTE. RAINet users are responsible for all the special customization to the router software. The UNIX workstations tend to use either *routed* or *gated* for routing (RIP), and SMAIL for the mail delivery system. RAINet has one primary DNS server, one secondary server, and several cache-only and resolver systems. Since most of the RAINet connections use standard phone lines, the auto redialer function of KA9Q is heavily used, so if a connection is broken, one end can call and reconnect to the other end. One of the customizations to PC/ROUTE was the addition of this feature. Also, *ripmerge* is used as much as possible to cut down the size of a RIP.

RAINet has a class B address space and uses a `0xfffffe0` netmask. Since each end of a SLIP connection requires its own subnet, plus the subnet for the group of hosts at the site, a class C network was not large enough. We figured that 30 usable hosts per site was plenty, and if there was a need for more, the site was probably too big for RAINet anyway and it should get its own class C address.

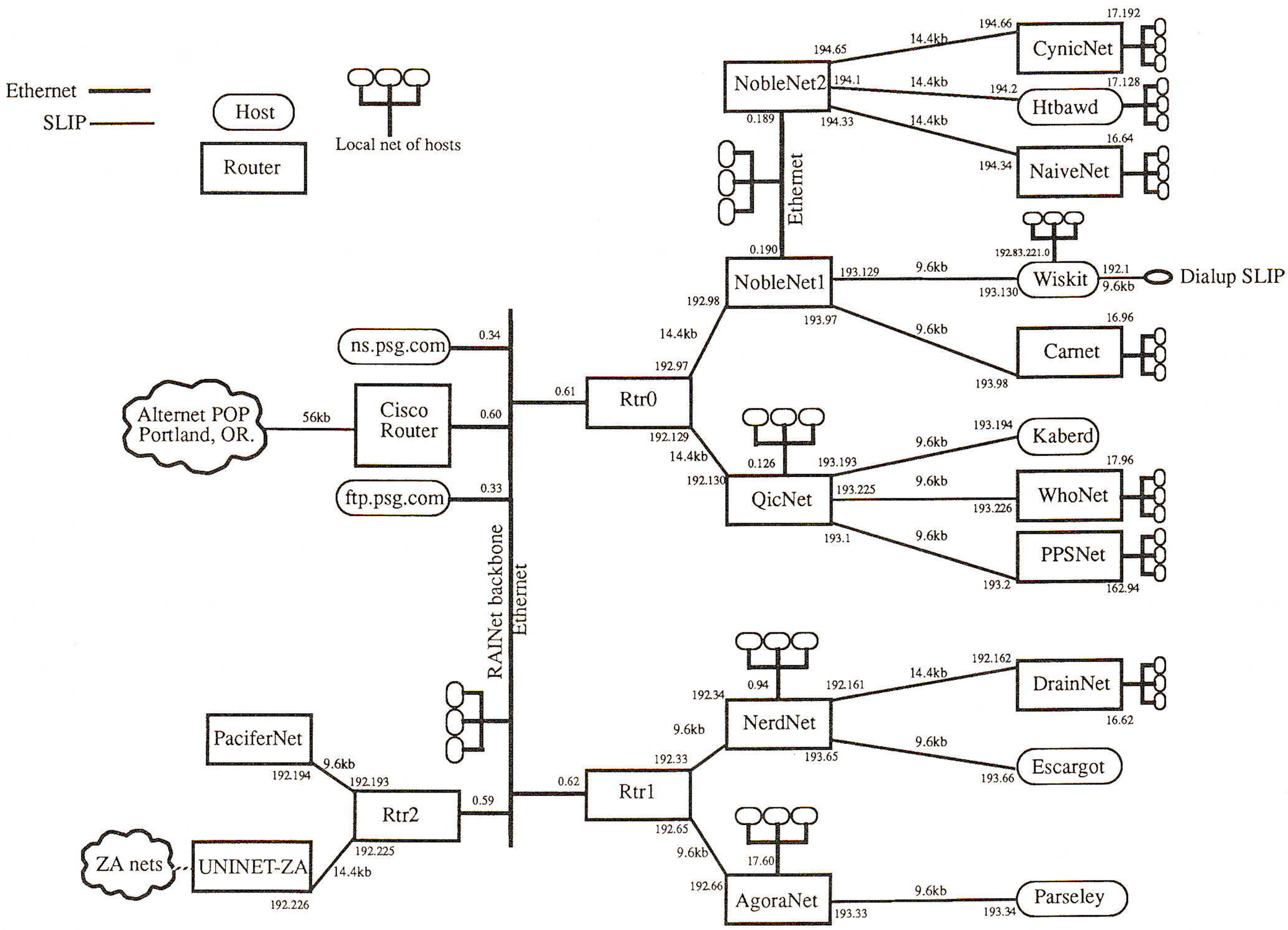



Figure 1: RAINet Configuration [147.28.0.0] March 1992.

**Economics**

Because of our current funding and volunteer work force, there is no charge to any site or user for using RAINet.

If an existing site with UUCP connections wishes to become part of `.rain.com`, but not RAINet, a simple application is all that is required.

*continued on next page*

## Profile: RAINet *(continued)*

If a site wishes to become part of RAINet, another short application is required (this is similar to the existing NIC form). In addition, the site must go through the follow steps:

1) Arrange with an existing RAINet site with an open slot for the connection. Generally each RAINet site limits their downward connections to three. Along with their upstream connection, this totals the number of serial connections a single 80286/80386 router can handle. A site can handle more than one router, and one of the firewall sites is already doing that.

2) Install residential phone lines at both sites. Since RAINet is using dialup SLIP but keeping it up once the connection is made, a dedicated phone line for both the site wishing to hook up and the site where the other router is located must be purchased. In Portland, the cost of a new phone line including installation is between $50–$75 dollars, and the monthly charge is under $25/month for each line.

3) Purchase modems for the two sites. The RAINet policy on hardware is that a site wishing to hook up to RAINet is responsible for purchasing all the hardware needed for the hookup. While V.32 and V.32bis modems are preferable, at the beginning of RAINet several sites were running SLIP over 1200 and 2400bps modems. A V.32bis modem such as a Telebit T3000 or Intel 14.4EX running with a DTE speed of 57.6Kbps can achieve 3Kb/s with FTP. V.32 modems can be purchased for under $400. V.32bis modems are slightly more expensive. The basic difference between a V.32 and V.32bis modem is that the modulation rate of V.32 is 9.6Kbps while V.32bis is 14.4Kbps. This shows up as both faster throughput and a better ability to handle simultaneous traffic with the V.32bis.

4) A site has two choices on where to hook their modem. Some sites choose to use a host-based SLIP, which means that one of the computer's serial ports is dedicated to the RAINet connection. The modem is plugged into the serial port and the SLIP (or someday PPP) is run on that port. No external router is needed. Host-based machines do not make good routers for other sites to connect to, since if the workstation crashes or is taken down for backups, the sites below are effectively cut off. RAINet encourages those sites who may at some point be taking on other connections themselves to go with a PC-based router. A 12Mhz+ 80286 or 80386SX monochrome 640K single-floppy system with Ethernet card and serial port (16550 UART) is all that is needed. The PC-router can run either KA9Q (which supports PPP) or PC/ROUTE. A complete PC-router can be purchased for less that $500, including Ethernet card and serial ports. The PC-router is hooked up to the site Ethernet. A typical PC-router can handle one Ethernet and four serial ports. As long as the PC-router stays up, the downstream sites are unaffected. All of the RAINet firewall sites use PC-routers, and most of the downstream sites use routers as well. But for a single machine site with no downstream connections, host-based SLIP is perfect.

**Network services**

Currently RAIN consists of two services: `.rain.com` and RAINet `.rain.com` is a registered domain name under which mail sites wishing to leave behind the pseudo `.UUCP` domain may register. The name space is allocated as simple nodes: (`sitename.rain.com`) or using sub-domain allocation for sites or organizations with multiple systems: (`sitenames.subd.rain.com`). In the latter case, a sub-domain administrator is responsible for allocating the sub-domain name-space and coordinating with the `.rain.com` administrator(s).

Administrators at sites wishing to register under `.rain.com` fill out a simple application similar to the regular zone registry form used by the NIC.

RAINet is the IP network using a class B address space allocated by the NIC. IP connectivity offers the typical services:

• *Electronic Mail:* Sites directly on the Internet use DNS for hostname resolution and deliver e-mail directly to the host or the designated mail forwarder. Sites below the firewall direct their non-RAINet e-mail to the closest firewall site for delivery.

• *Remote Access:* Several users on both RAINet firewall sites and downstream sites need Internet access to carry out research and/or school projects. A few accounts for non-firewall-site users have been set up to allow Internet access both from RAINet to the outside and visa-versa. For example, Portland Public Schools use one of the firewall sites to access other Internet sites and exchange research in the area of psychometrics. Another proposed experiment will allow students at a local high school to access Internet resources listed in the various Internet cookbooks such as "Zen and the Art of the Internet."

• *News:* RAINet gets an NNTP feed from UUnet. The main news site in RAINet in turns feeds news to several other RAINet machines as well as several other Portland area systems. Local area groups, including a RAINet hierarchy, are passed around in a triangle with Oregon Graduate Institute and Tektronix. The other RAINet sites feed several Portland area systems via UUCP. The total news traffic in and around RAINet is several hundred megabytes per day. In addition, RAINet sites offer news-reading accounts for several hundred users.

• *File Transfer:* Though file transfer at V.32/V.32bis speeds is not the greatest, FTP is still a popular use of RAINet. Some of the firewall sites have set up anonymous FTP areas for use by the Internet community.

• *Other:* Several users are experimenting with such things as WAIS, IRC, MUDs, Gopher, and various proposed RFCs (e.g., RFC 931).

**Growth**

RAINet is currently adding three to five `.rain.com` sites a month and one to two IP connected sites per month. Our sub-netted class B network allows for over 4000 32-host nets, so we have plenty of address space. As we add sites, our hierarchy grows. In order to keep the number of hop counts down, we may need to expand sideways by adding more routers to the Ethernets of upstream sites. This is not a problem now, but if the growth rate continues, the hop-count problem will be a definite consideration.

**Communication with other toasternets**

The term "toasternet" came into popular use as a description of networks that have a multitude of devices hooked up to it, including in the case of one demonstration at a recent INTEROP conference, an Ethernet equipped toaster. We use the term "toasternet" to convey the multitude of systems in RAINet, and the various pieces of hardware and interconnect technology used to hook these systems together.

It is our hope that low-cost IP "toasternets" continue to spring up all over. With more companies providing IP connectivity options, we feel such low-cost networks are ideal for the next generation of Internet inhabitants, such as public schools, libraries, hobbyists, and computer clubs, to join the global electronic community.

*continued on next page*

## Profile: RAINet *(continued)*

**Mailing list**

A mailing list for toasternets is available:

```
toaster-request@psg.com        ; for list additions/deletions
toaster-list@psg.com           ; mailing list reflector
```

We are in contact with several other individuals and groups who either already have toasternets running or are in the planning or implementation stages.

**Future plans**

RAINet hopes to continue to examine alternative connection topologies, in terms of both higher reliability and lower cost, as well as the simple research aspects of discovering and perfecting new methods of hooking computers together. We also want to expand the use of leased lines instead of regular residential phone lines, as they are both less expensive and more reliable. As soon as is feasible, we will move from SLIP and CSLIP to PPP. Several of the sites running host-based SLIP are unable to run host-based PPP because it is not available. In addition, some of the PC-based routers in RAINet are running PC/ROUTE rather than our modified KA9Q. PC/ROUTE does not support PPP. We will continue to look at other alternative protocols such as POP and IMAP for supporting simple (MS-DOS) mail-only clients.

We will also continue to add new .RAIN.COM and RAINet sites. We are actively working on expanding our public school connections both within the Portland Public Schools, and with other local school districts. In addition, we are in contact with the county library system in hopes of getting them connected to RAINet as well.

Only a few of the sites have workstations and/or routers that have SNMP agents. We are interested in using SNMP-based products for both trouble-shooting and network load management. We would like to expand the number of sites involved with SNMP.

RAINet continues to discover new and wonderful things about community networking. We hope these discoveries never stop.

**Contacting the authors**


| | |
|---|---|
| Steve Neighorn is at: | neighorn@qiclab.scn.rain.com |
| Randy Bush is at: | randy@psg.com |
| Jeff Beadles is at: | jeff@onion.rain.com |
| RAINet administration: | rain-admin@rain.com |


***Attend the Toasternet BOF on Thursday, May 21 at 7:30pm.***

**STEVE NEIGHORN** received is B.S. in computer science from Portland State University in 1985, and is working feverishly on his Master's thesis on distributed virtual memory and load balancing at Oregon Graduate Institute (OGI). He worked his way through college at Portland Public Schools in the Research and Evaluation Department. He also spent one year at Intel Corporation and three years at Sun Microsystems. He is now dividing his time between independent consulting at Intel Supercomputer Systems Division, thesis research at OGI, and various RAINet projects, including K–12 internet hookups, in the Portland-metro area.

**RANDY BUSH** is Director of Enginering at Servio Corp, the leading commercial developer of OO databases and tools. He has been a user and occasional implementor of networking in the US from the '70s to the current day Internet, UUCP, and FidoNet. He has been involved in in integration of appropriate networking technology in Southern Africa for over three years. He sketched out and helped deploy a multi-country (South Africa, Botswana, Namibia, Zimbabwe, and soon others) network. Randy is a designer and the technical coordinator of the Portland Oregon IP metronet, RAINet.

**JEFF BEADLES** has worked with the UNIX operating system for the past 11 years as a user, administrator, and programmer. He worked at Tektronix for 8 years on various UNIX workstation products, focusing on IP networking. By day, he works for Mentor Graphics developing a software development environment for multiple UNIX platforms. By night, he is the security contact for RAINet, provides Internet-like services to those within RAINet, and does research on new possiblilties for information exchange.

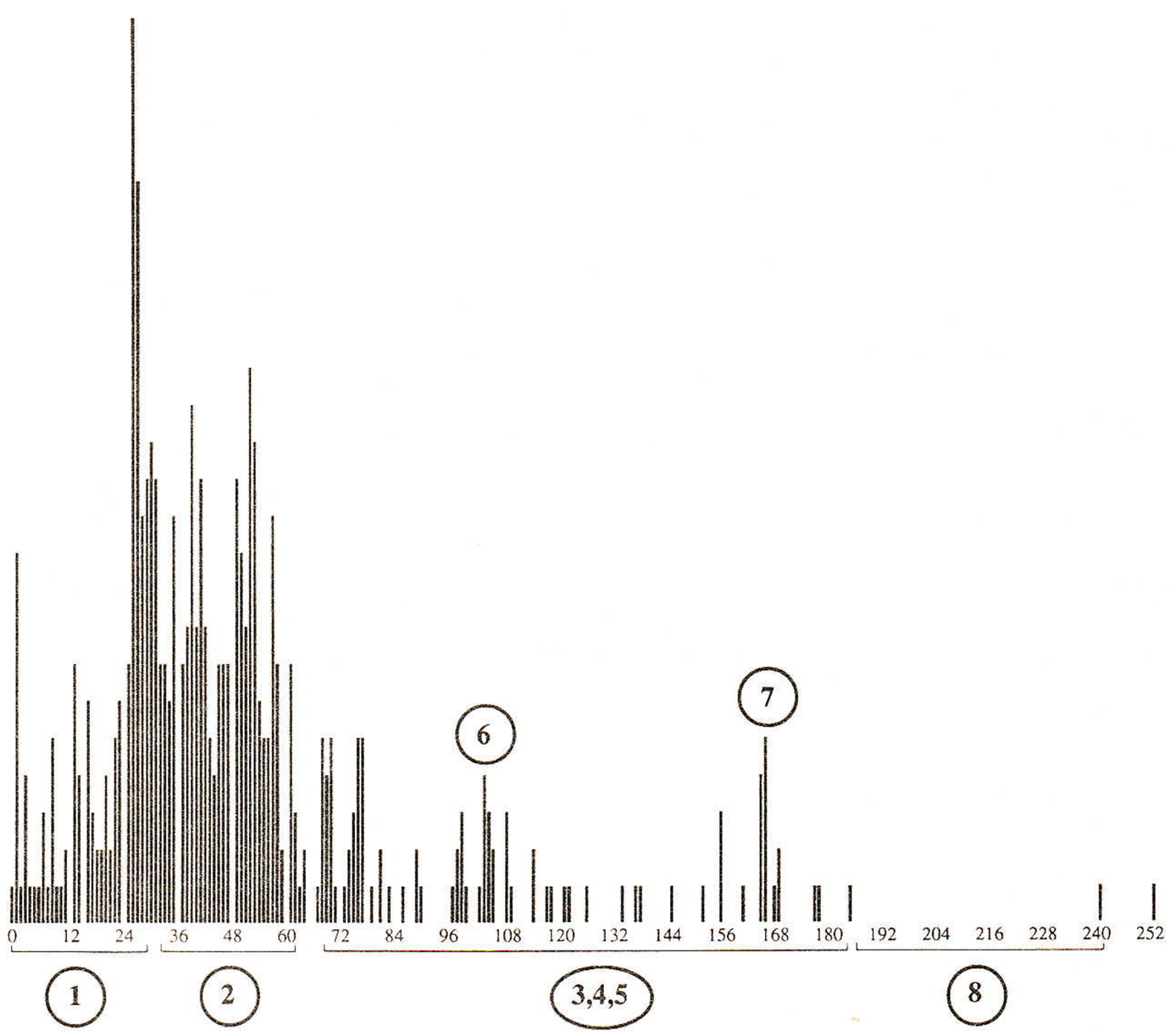
## One way of Measuring Internet Growth

by Jon Crowcroft, University College London

It has been observed that the Internet is growing fast; *very* fast. So fast that even a Ferrari with a trunk full of magnetic tapes is totally unimpressive along side it.

**Fast**

How fast is that? Well, if you stand at the end of a long hall with your CD Remote Control in hand, and the TCP/IP CD ROM on the player at the other end, write a few awk-ward words, light the blue touch paper, stand back in awe, and press "play," you find the following:




*RFCs published per month: April 1969–September 1991.*

Here we have annotated various periods of intense interest, and drawn several casual inferences from them:

1. The RFC editor worked very hard to finish his Ph.D. as did many other assiduous US workers.

2. Internet congestive collapse was caused by too many research groups exchanging research notes about congestion avoidance algorithms in an attempt to get: a) more DARPA funding, b) more NSFNET bandwidth, c) more Gore blimey.

3, 4, 5. The IETF meetings start happening more regularly.

6. A common virus lays many people low.

7. Columbus discovers *PostScript*.

8. The IETF discovers *compress*.

**What can we deduce from this?**

It is obvious that most of the traffic is caused by people writing RFCs. It is also the case that since IETF meetings have been held regularly, hardly any traffic goes over the net any other time. Thus the entire Internet backbone should immediately be scrapped, and a system of trucks and tapes acquired, which together with a little more foresight, should prove adequate for the responsiveness now needed. Failing that, SMDS (*Seldom a Meltdown Demands a Sanding*) should prove popular. Dis-claimer is better than DAT.

# The Growing Internet

by Frank Solensky, Clearpoint Research Corporation

## Introduction

The question "How large is the Internet?" has always generated a great deal of interest. Some of this curiosity, of course, is that it is pleasing to compare how many more places one can connect with now as opposed to some earlier time—what the author thinks of as the "My, how big you've grown!" effect. This question is also of immediate interest to those who are planning for the future: someone who is designing a new router will want an estimate of how many new networks will be created over the next several years before deciding how much memory should be put into the system.

## Linear regression

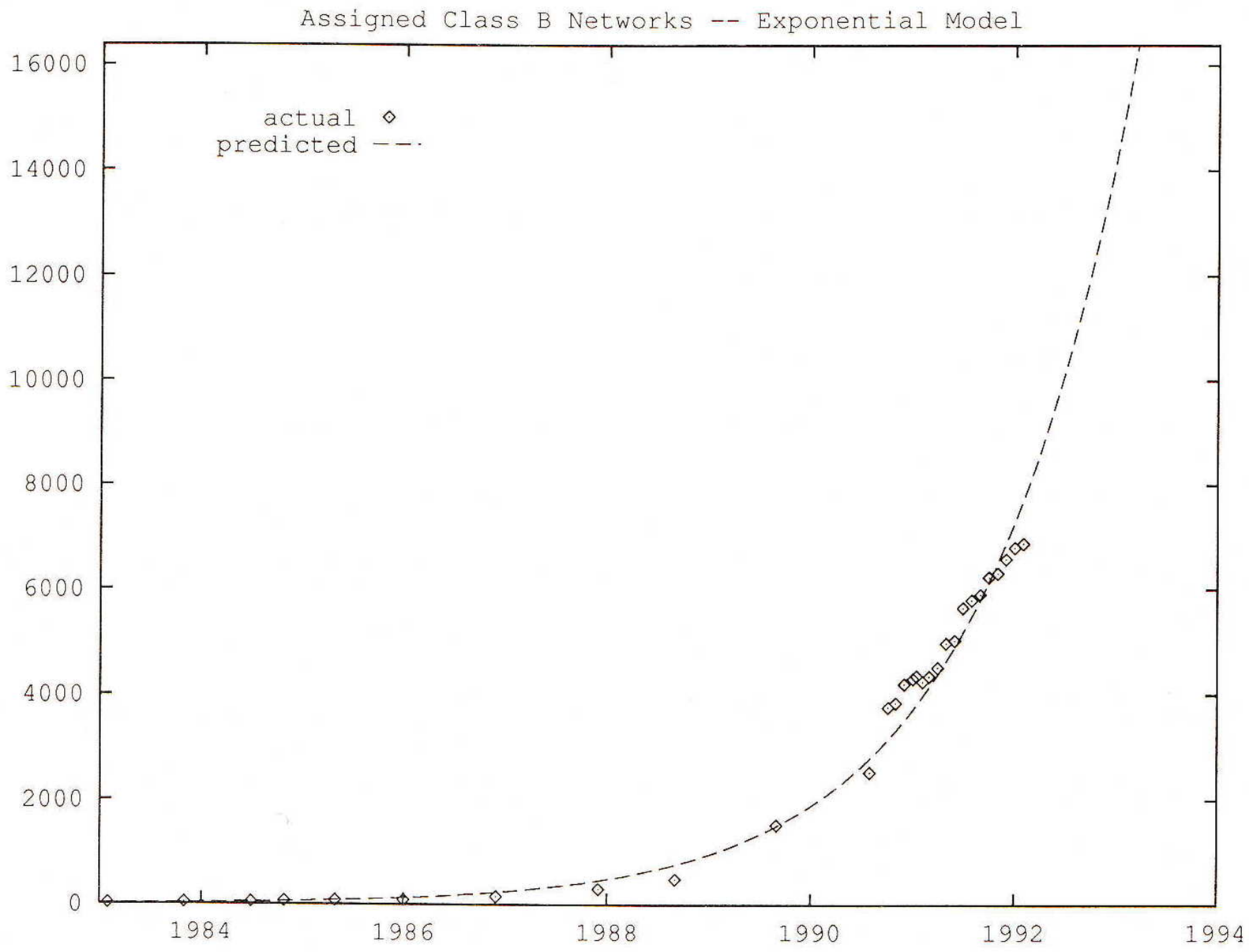
A simple approach to this problem would be to make an estimate of the annual growth rate. This is accomplished by transforming the data into their logarithms and performing a linear regression on the new data points. In effect, this consists of finding the straight line that best fits the data when the y axis is on a log scale. Reversing the original transformation creates a model of the form:

$$y = e^{(A\,t + B)}$$

The computed value of "A" then represents the average growth rate during the unit time interval of "t."

Using this approach and the most recent data available, we find that:

- The assignment of IP network numbers has been growing at a rate of about 36.7% per year.
- The growth rate of Class B network number assignments alone has been averaging 67.4% per year. If continued, this would result in the depletion of assignable Class B numbers on March 14, 1993 at 7:48pm.
- The number of networks that appear in the NSFNET policy routing databases has been increasing at a rate of 77.4% per year.



**Problem**

There is a fundamental problem in assuming a constant growth rate, however: extrapolating it into the indefinite future can generate results that are, at best, misleading. By way of example, if we were to take the attendance figures for INTEROP and assume that its average annual growth rate of 155.4% could be sustained, one would have to plan for about 5.3 million people attending in the fall of 1996—approximately equal to the current population of the San Francisco–Oakland–San Jose metropolitan area!

**Logistic curve**

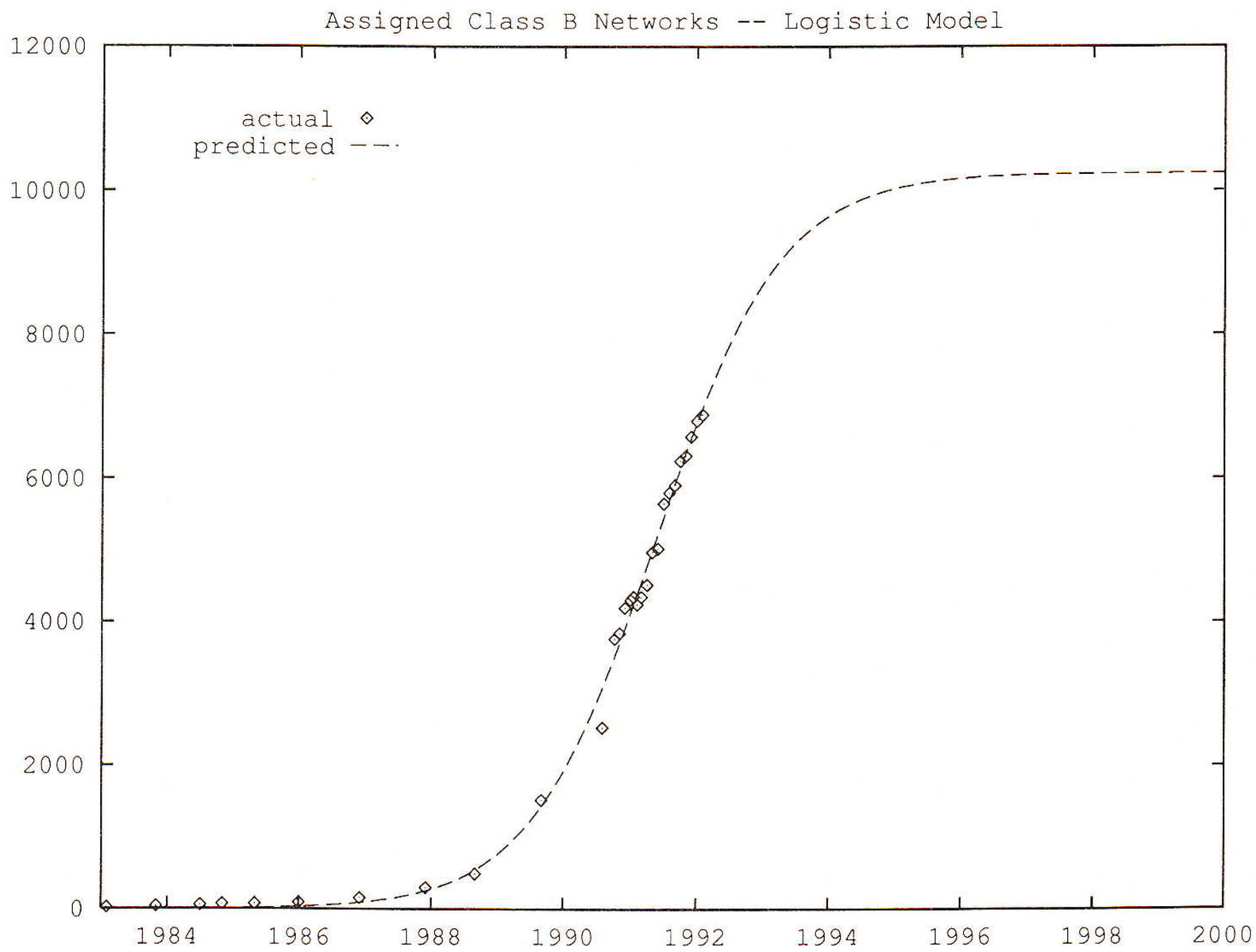
For this reason, we turn to the *logistic curve* for a more realistic analysis. This curve was first derived by the Belgian demographer Pierre-François Verhulst who, in 1845, used it to predict the growth of the population of the United States. More recently, it was used in an analysis of the growth rate of BITNET [1]. It takes the form:

$$\frac{dP}{dt} = k\,P(M - P)$$

or, in English, the change in the population (P) is proportional (k) to the product of its current level and the amount of potential growth left in the system (M – P). The value M is the upper bound of this population. This equation can be transformed into:

$$P_t = \frac{M\,P_0}{P_0 + (M - P_0)\,e^{-kMt}}$$

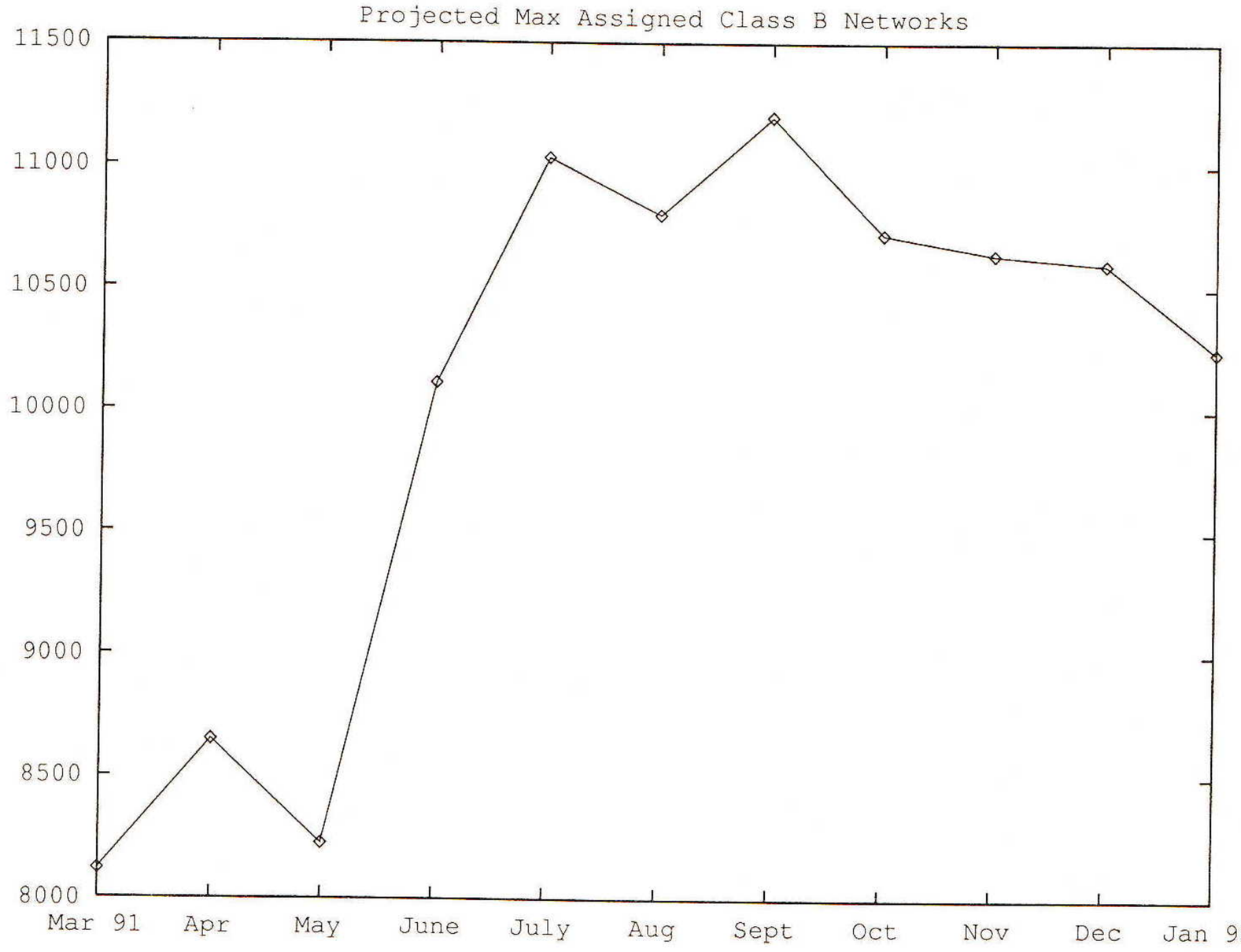
and results in an "s"-shaped curve that rises slowly at first when *t* is close to zero, then more rapidly in the middle and flattens out just below the upper bound of *M* as *t* continues to increase.




*continued on next page*

## The Growing Internet *(continued)*

We can then use nonlinear regression techniques to estimate the parameters to this equation. The exact methodologies used are beyond the scope of this article; textbooks such as [2] are good sources of this information. Focusing on the assignment of Class B network numbers for the moment, we find that this population currently levels off at about 10,250 networks (see graph).




While predictions have been fluctuating within the range of 10,000 to 11,500 Class B networks over the last several months, it bears noting that this offers no guarantees. Using data through May 1991, the predicted maximum number of Class B networks would have been 8,227 networks. As of this writing, 6,883 Class B network numbers have been assigned. Therefore, if we have gone past the inflection point of the curve—the point at which growth starts to slow as the population reaches its maximum value—the fact that this predicted upper limit is itself prone to revision as more recent data becomes available demonstrates that we cannot be very far beyond it.

### References

[1] Vijay Gurbaxani, "Diffusion in Computing Networks: the Case of BITNET," *Communications of the ACM,* December 1990.

[2] G. A. F. Seber & C. J. Wild, *Nonlinear Regression,* John Wiley & Sons, 1989.

**FRANK T. SOLENSKY** received his B.A. (1977) and M.S. (1979) from New York University. He has been working with communications networks and protocols for most of the last seven years. He has been with Clearpoint Research Corporation for just over a year where he has been primarily responsible for the development of the company's first router product and really thinks the Red Sox have a chance to win the series this year. But he's thought that for most of the last seven years, too.

# How Did We Get 727,000 Hosts?

by April Marine, SRI International

**Introduction**

The question SRI answers most these days is, "How do I join the Internet?" For those people asking that question, who are just now getting caught up in the current internetworking world of multi-protocol, multi-vendor, multi-media multi-mania, it's almost impossible to imagine what the Internet of today was like when it started.

**RFC 1**

Not many people these days have a copy of *Request For Comments No. 1* (RFC 1). In fact, other than the Postel Personal Archive Collection reportedly secreted away in Jon Postel's garage, I'm unaware of any place, other than SRI, that has a copy of every RFC ever issued. If you did happen upon a copy of RFC 1, you'd notice it contains this puzzling open question: "The HOST is supposed to be able to send messages to its IMP. How does it do this?"

Well, they obviously found an answer between April and October of 1969 because we find plans for sending test messages on the new network in RFC 4. To help clarify this testing scheme, the RFC includes the following helpful network configuration diagram:

```
SRI|
   |
   |
   |
UCLA|
```

At that time, there were four hosts on the network: SRI, UCLA, UCSB, and Utah. (There's the answer to a trivia question SRI is asked about once a year.)

So, how did we get from these four hosts to the huge Internet of today? I think Steve Crocker had the answer in the first paragraph of RFC 3:

> "The Network Working Group seems to consist of Steve Carr at Utah, Jeff Rulifson and Bill Duvall at SRI, and Steve Crocker and Gerard Deloche at UCLA. Membership is not closed."

**Growth**

That "membership is not closed" is what made the Internet of today. In January 1992, there were 727,000 hosts on the Internet. The leap from the simple four-host-network diagrams in RFC 4 to the logarithmic growth charts of RFC 1296 is mind-boggling. SRI, the site of one of first hosts, and the site of the first of today's family of network information centers, has had a privileged position from which to witness the birth and growth of what has become a worldwide community. In fact, SRI has an extensive archive of Internet history, some of which was used for data reported in RFC 1296, "Internet Growth (1981–1991)." Let's look at a few numbers from that RFC.

In the days before the *Domain Name System* (DNS), which is only about ten years old, SRI maintained a master host table that listed every host on the network. In August 1981, this table contained listings for a total of 213 hosts. So, in the twelve years from 1969 to 1981, the network grew by only 209 hosts. Think about it—all the hosts on the ARPANET ten years ago would fit on one Class C network of today.

*continued on next page*

## How Did We Get 727,000 Hosts? *(continued)*

**Four factors**

The natural question someone new to the Internet would have at this point is, "What happened in the last ten years that made the Internet grow so big so fast?" People don't have a lot of time these days for history, so let's have the quick and dirty answer. Simply put, there were four factors: the birth of TCP/IP, the government decree to use it, the *Domain Name System* (DNS), and the evolution of affordable computers. The numbers support this view.

Between August 1983 and October 1984, the network grew from 562 hosts to 1,024, which was quite a significant increase given the previous slow rate of growth. Why? Because in 1984 the ARPANET split into two networks, the MILNET and the ARPANET, and the government decreed that all their networks must use the TCP/IP protocols. In other words, the government decided that the network had reached a point where it could be used for day-to-day operations in addition to its original role as a testbed for the development of network protocols. From that point, the network grew steadily, so that in November 1986, there were 5,089 hosts.

One year later, in December 1987, there were 28,174 hosts. Hello! That's a big jump. During this time, the DNS was implemented. The DNS is a a hierarchical system which allows distributed management of host naming and addressing. Its implementation freed the network from depending on a centralized host table (much to SRI's relief!) and delegated the power to update much of the host information in the DNS to local administrators. Obviously, they were more than ready to assume the responsibility.

Just one more paragraph of numbers. In January 1989, there were 80,000 hosts on the Internet. In January 1991, 376,000. This January, there were 727,000 hosts. Again, why so big so fast? There are a lot of factors here, including the National Science Foundation's support of the NSFNET, the trend for mid-level networks to bring networking out to the communities, the momentum and enthusiasm of having more people use networking to improve networking, and the increasing importance of the commercialization and privatization movements. Most significant, perhaps, is the decrease in the cost of computers, which led to there being a lot more of them around, which led naturally to the desire to hook them all up.

**ZONE**

Now you're probably thinking, "Sure, these are nice numbers, but where did they come from? How do we really know there are 727,000 hosts on the Internet today?" SRI has a program called *ZONE*. Details of what ZONE is, and what it searches for, are explained in RFC 1296, but basically it's a program that walks through the DNS and finds out what's there. To answer the most common question about ZONE: Although ZONE does not count MX records, it may count hosts that are only accessible from the Internet via electronic mail because the addresses of these hosts are advertised in the DNS even though their sites filter the traffic that can get to them.

**Conclusion**

Will the Internet continue to grow in leaps and bounds? Sure. People need to communicate and need to get information, and the Internet has become an essential tool to get their jobs done. More and more people every day are opening the door and stepping out into the world of the Internet. And there are lots of places, including SRI, that will help you find that door and open it. In fact, SRI has just written a new guide called "Internet: Getting Started" that explains how to join the Internet and gives you the basics of what it all means. Contact SRI at 415-859-6387 for details.

It seems fitting to end as we started, with a quote from RFC 1:

> "The above described primitives suggest how a user can make simple use of a remote facility. They shed no light on how much more intricate use of the network is to be carried out."

Seems like they're still working on that one, no?

**References**


[1] Clark, D. D., Chapin, L. A., Cerf, V. G., Braden, R. T., Hobby, R., "Towards the future Internet architecture," RFC 1287, December 1991.

[2] Crocker, S. D., "Host software," RFC 1, April 1969.

[3] Crocker, S. D., "Documentation conventions, RFC 3, April 1969.

[4] Lottor, M., "Internet Growth (1981–1991), RFC 1296, January 1992.

[5] Shapiro, E. B., "Network timetable," RFC 4, March 1969.

[6] Dern, D., "The ARPANET is Twenty," *ConneXions,* Volume 3, No. 10, October 1989.

[7] Mockapetris, P., "Introducing Domains," *ConneXions,* Volume 1, No. 6, October 1987.

[8] Mockapetris, P., "Domain Names: Concepts and Facilities," RFC 1034, 1987.

[9] NNSC Staff, "Profile: NSFNET," *ConneXions,* Volume 1, No. 2, June 1987.

[10] Braun, H-W., "The new NSFNET backbone network," *ConneXions,* Volume 2, No. 12, December 1988.

[11] Roberts, M., "The NREN Takes Shape," *ConneXions,* Volume 5, No. 6, June 1991.

[12] Roberts, M., "NREN Bill Signed into Law," *ConneXions,* Volume 6, No. 2, February 1992.

[13] Barron, Billy, "Another use of the Internet: Libraries Online Catalogs," *ConneXions,* Volume 5, No. 7, July 1991.

[14] Quarterman, John, "Networks: From Technology to Community," *ConneXions,* Volume 5, No. 7, July 1991.

[15] Schwartz, Michael F., "Resource Discovery and Related Research at the University of Colorado," *ConneXions,* Volume 5, No. 5, May 1991.

[16] Kahle, Brewster, "An Information System for Corporate Users: Wide Area Information Servers," *ConneXions,* Volume 5, No. 11, November, 1991.

[17] Deutsch, P. & Emtage, A., "The *archie* System: An Internet Electronic Directory Service," *ConneXions,* Volume 6, No. 2, February 1992.


**APRIL MARINE** joined SRI International's Network Information Systems Center more than six years ago when it was the site of the Network Information Center. She currently works with the SRI team that provides Internet information services, is active in the IETF User Services Area, and is the editor and co-author of the guide "Internet: Getting Started."

# An Introduction to the ISODE Consortium

by Steve E. Hardcastle-Kille

**Abstract**

The ISODE Consortium has been founded to support further development of ISODE as a package on which vendors can build OSI products and as a package which continues to be used in the research community. The Consortium will provide the vendor-neutral architectural and administrative leadership that is required to make this work. The ISODE Consortium will aggressively evolve ISODE, with particular focus on Directory Services and Message Handling Services.

The ISODE Consortium is a not-for-profit organisation, which is self supporting through membership fees and product royalties. Membership is open to any organisation in any country. In addition, there is a special class of non-voting Individual Membership.

This article gives a brief overview of why the ISODE Consortium is needed, how it will be set up and organised, and its technical goals.

## ISODE

The ISO Development Environment (ISODE) is an implementation of selected OSI protocols and applications, which runs on a wide range of UNIX and UNIX-like operating systems. The key components of ISODE are:

- Connection Oriented Transport Service (COTS) over vendor stacks:
  - TP4/CLNS
  - X.25 or CONS
  - TCP (using RFC 1006)
- Transport switch to allow simultaneous working over multiple stacks.
- The Connection Oriented OSI upper layers: Session and Presentation.
- Selected *Application Service Elements* (ASEs): Association Control (ACSE); Remote Operations (ROS); Reliable Transfer (RTS).
- OSI *File Transfer Access and Management* (FTAM): protocol libraries; server and responder.
- OSI *Directory Service* (*QUIPU*): DSA and selected DUAs.
- *Message Handling Services* (*PP*): A Message Transfer Agent, suited to volume switching and protocol conversion.

Over a hundred people and organisations have contributed to ISODE, and there is not space to credit all of them here. ISODE was originated by Marshall T. Rose, and the current source tree is managed by Julian Onions and Colin Robbins. Organisations which deserve special mention are: The UK Joint Network Team (for providing funding); Northrop Corporation; The Wollongong Group; University College London; University of Nottingham; MITRE Corporation; X-Tel Services; Brunel University; CSIRO; and the University of Michigan.

## ISODE successes

ISODE was originally developed as a "development environment" to experiment with the OSI upper layers. The original ISODE was "Openly Available," which is effectively public domain, and spread rapidly through the research community and into many commercial organisations. It has been used for a very wide range of work, and is the most widely deployed OSI implementation. It has succeeded impressively in meeting its original goals.

Although the original ISODE was targeted primarily at programmers, the largest uptake has been by organisations wishing to use ISODE applications to run services. This is particularly true of the X.400 and X.500 components, but also FTAM to a lesser extent. The *PP* X.400 MTA is being used at an increasing number of mail gateways for its protocol conversion facilities between RFC 822 and X.400 and its suitability for high volume message switching. It is also being adopted by an increasing number of organisations as a "mail hub," due to its advanced management facilities.

The *QUIPU* X.500 implementation has been closely tied in with the pilot deployment of X.500 in the research community. This pilot now encompasses over half a million entries in over four hundred DSAs (*Directory System Agents*) in twenty countries. This includes the PSI White Pages Project in the US, and the PARADISE Project in Europe. QUIPU is the dominant DSA implementation used for this piloting, and QUIPU based DUAs are used extensively.

**The ISODE Consortium**

The ISODE Consortium is being set up in response to the success of ISODE, and is intended to give ISODE a firmer base so that it can continue to evolve and thrive. The ISODE Consortium intends to build on the successes of the current ISODE, and to address the weaknesses. The Consortium will evolve ISODE to make it more attractive and straightforward to build products upon, by tackling issues which could not effectively be met with the earlier setup. By focusing on system components, such as layer services and MTAs, the ISODE Consortium will allow vendors to take building blocks, which are complex and costly to build, and focus efforts on user interfaces to provide product differentiation.

The cooperative approach being facilitated by the ISODE Consortium will allow for more cost effective development than would be possible for a vendor working alone. The Consortium will take ISODE out of the public domain, and commercial organisations will gain access to the source by joining the ISODE Consortium. This will later provide the ISODE Consortium with revenue from product royalties, and thus ensure the long term viability of the ISODE Consortium. The Consortium will also retain and strengthen links with the research community. The ISODE Consortium will make ISODE available to academic organisations and government or not-for-profit organisations with research as their primary purpose by means of a simply administered zero cost licence. The Consortium will also work closely with organisations such as RARE and IETF.

**Product focus**

The major goal of the consortium is to evolve high functionality OSI-based applications, which can be used directly and as a basis for products. The development will be done by a mixture of direct work by the Consortium, subcontract, and contribution from members. This combination will provide rapid and effective technical evolution. The key components which the ISODE Consortium will work on will be application level components of networked applications. The major focus will be on "system" components such as MTAs (*Message Transfer Agents*) and DSAs. These complex components are a major strength of ISODE, and a fundamental component of any product. There is relatively low potential for value added function compared to user interfaces, and so it is very attractive for vendors to buy in this technology. This split means that APIs will be critical. All APIs used by the ISODE Consortium will be clearly defined, and will in general be publicly available.

*continued on next page*

## Introduction to the ISODE Consortium *(continued)*

Work will be done in a way which retains independence of lower layers, and the ability to use multiple stacks. Support for use over TCP/IP will be promoted in parallel with use over OSI lower layers.

Message Handling Services and Directory Services will be the main focus of the ISODE Consortium, particularly in the first few years. They are seen both as useful end applications and as key building blocks, and the areas discussed below will be particularly focused onto these applications. Tracking standards, and conformance will be fundamental, as well as adding in new functionality and improvements. The OSI protocols will be a core component, but the overall system will include components for interoperability with non-OSI systems and private extensions where appropriate. A major change to the X.500 DSA will be to define a database API, so that new databases and mappings onto proprietary databases may be offered as value-added products.

**Technical direction**

- *Conformance:* There has been substantial work on interoperability testing with the ISODE components, but almost no work on conformance testing. Conformance testing, and the specification of PICS to define conformance levels will be a key aspect of early ISODE Consortium work.
- *Configurability:* Whilst the current components provide high functionality, they are too complex to configure, and there will be a transition to "plug and play."
- *Management:* Application management is a key problem, and the ISODE Consortium strategy has three basic components: The X.500 Directory will be used extensively; The industry standard SNMP will be used for overview monitoring of large numbers of components; Application specific management will be used for some purposes.
- *Security:* Addition of effective security to the ISODE Applications is an important medium term goal. Use of X.509 public-key based security is the technology most likely to be used.
- *Performance:* Whilst broadly in line with performance of similar products, improving the performance of ISODE will be a high priority. Three complementary areas will be worked on: ASN.1 compilers; Careful tuning of the full OSI Stack for the key applications; Use of a lightweight stack, with mappings onto COTS, CLTS, TCP, and UDP.
- *Integrating New Applications:* The issue of introducing new technology is important. In general, this will be done by a request for technology. X.400 Message Store is a likely candidate for early addition.

**Membership**

The consortium will raise revenue primarily by membership fees and product royalties. There are several types of membership, which are summarised here:

- *Commercial:* This is for commercial organisations, who may be vendors or users of OSI products. Membership rate is determined by the annual revenue of the organisation.
- *Research:* This is for Academic/Educational organisations, and for not-for-profit and government organisations with research as their primary purpose. There are two rates, with the lower being for academic institutions, and organisations with less than 150 members.

- *Non-Commercial:* This is for not-for-profit and government organisations which do not fit into the research category. There are four rates of membership, based on the size of the organisation.
- *User:* This is for organisations that will gain access to ISODE by using products from ISODE Consortium members.
- *Research Consortia:* This is for organisations which are consortia which support research organisations. Research networks are typical members of this type. There are three rates of membership, determined by consortium size.
- *Individual:* A low cost membership for individuals.

**Organisation**

The ISODE Consortium has been established as a professional non-profit corporation (US 501.c(3)). This initial startup was achieved by funding from MCC (*Microelectronics and Computer Technology Corporation*), which is a research consortium of primarily US companies. MCC's objective is to act as a technology transfer agent between academic research and industry.

The president of the ISODE Consortium is Steve Hardcastle-Kille, who is also currently a Senior Research Fellow at University College London, and has played a key role in the design and evolution of ISODE. This will be a full-time position when full operation starts. The ISODE Consortium will establish a US office in May 1992 in the Washington, DC area. The Consortium will have a strong presence in Europe, and startup of the European office is planned for early 1993.

The initial board of directors of the ISODE Consortium is: Phil Cannata (MCC); Lyman Chapin (BBN); Dave Farber (University of Pennsylvania); Dan Lynch (Interop); Hugh Smith (X-Tel Services). The board will be elected by the membership. Members will participate in the Policy Steering Committee and Technical Steering Group to determine strategic and tactical direction for the ISODE Consortium.

**Contact information**

| | *US Office* | *European Office* |
|---|---|---|
| | ISODE Consortium<br>c/o MCC<br>P.O. Box 200195<br>Austin, TX 78720<br>USA | ISODE Consortium<br>P.O. Box 505<br>London<br>SW11 1DX<br>UK |
| Voice: | +1 512-338-3340 | +44 71-223-4062 |
| Fax: | +1 512-338-3600 | +44 71-223-3846 |
| E-mail: | `ic-info@isode.com` | `S=ic-info;`<br>`O=ISODE Consortium;`<br>`PRMD=ISODE; ADMD=0; C=GB;` |

**STEVE HARDCASTLE-KILLE** is the president of the ISODE Consortium. Prior to this, he was a Senior Research Fellow in the department of Computer Science at University College London. He has a BA and MA in Physics from Brasenose College, Oxford, an MScs in Electrical Engineering from University of Manchester Institute of Science and Technology and Stanford University. He has worked at University College for the past ten years, and has conducted research in a range of areas relating to networking, messaging, directories, and distributed systems. He is the architect and project manager for the *PP* (X.400) and *QUIPU* (X.500) systems.

***Hear more about this topic at INTEROP 92 Spring. Attend the ISODE Consortium BOF on Wednesday, May 20 at 7:30pm.***

# Profile: FARNET (The Federation of American Research Networks)

by Laura Breeden, FARNET Executive Director

**In the old days**

Many of you Internet old-timers (say, those of you born before 1960) have no doubt reflected that in the old days it was easy. There was one network (the ARPANET), one funding agency (DARPA), one network information center (SRI NIC), and one network operator (BBN). If you wanted a network connection, you went to DARPA and pleaded your case. That was that. Of course, along with this dazzlingly simple array of choices went a stern policy on network use and a limited set of applications, which you could display on a terminal whose "graphical user interface" relied heavily on the asterisk (*) for its effect.

**Today**

That was then, this is now. Today there are providers of Internet access operating in every state in the US and 38 foreign countries. The number of users of the Internet is estimated to be in the 2–4 million range, and the use of pop-up windows and pull-down menus to access network services is routine. The services include vast directories of publicly available files and programs, full-text databases stored on massively parallel computers, and OSI X.500 directory servers.

**FARNET**

FARNET is a non-profit association of network service providers and other organizations interested in the application of network technology to problems in research and education. Set up in 1986 as the Federation of American Research Networks, and incorporated in 1990, FARNET currently has 32 members (see list at the end of this article). They include state and regional networks, interexchange carriers, and national networks, among others.

FARNET answers a need that did not exist in the old days, B.C. (before competition, commercialization, choice, and confusion): the need for coordination and information-sharing on a regular basis across the multiple administrative boundaries between constituents of the Internet. It supports the development of a stronger and more usable network end-to-end by focusing the attention and resources of the members on common interests and problems.

**What does FARNET do?**

FARNET's primary goals are improving Internet user and information services, making the Internet more robust and easier to operate, and maximizing the effectiveness of FARNET member organizations. In pursuit of these goals, it organizes workshops and educational meetings, negotiates discounts on products and services, coordinates activities across multiple networks, and acts as an information conduit, linking the members with national policy-makers and NREN stakeholders. It also takes on projects of national interest or importance that fall outside the scope of any individual member organization. Perhaps actions speak louder than words. Since April 1991, FARNET has:

- Conducted policy workshops on the follow-on to the current NSFNET backbone contract and the improvement of end-to-end operational reliability in the Internet
- Brought together K–12 educators from across the country to share their experiences with school networking
- Met jointly with the IETF to strengthen cooperative activities
- Negotiated member discounts on dial-up Internet access services and low-cost IP routers
- Begun publication of an on-line newsletter

- Announced plans to respond to the NREN NIS solicitation as provider of coordination services and "user ombudsman"
- Secured grant funds from the National Science Foundation and ANS to support programs in user services, K–12, and understanding of network usage
- Met regularly with the Coordinating Committee on Intercontinental Research networks to discuss policy and planning concerns

**The network that will be**

If there is a single vision shared by all of FARNET's members, it is of the potential of networking to transform learning, research, and community affairs by bringing people together in new ways. State colleges and universities, public library systems, school districts, businesses, economic development agencies, and industrial and government research laboratories are being joined in an electronic information collaborative. Today, most of this information is textual, with a smattering of graphics. Within ten years, it will include voice and video. The members of FARNET are the Johnny Appleseeds of this technology, bringing new communities of users into the Internet, building tools, and nurturing new applications.

**For more information**

A complete directory of FARNET information can be browsed on Internet host `farnet.org`. FARNET's headquarters is in Waltham, MA at 100 Fifth Avenue (ZIP 02154). For more information, call 617-890-5120 or send e-mail to `breeden@farnet.org`.

**Member list (as of March 1992)**

*ANS* (Advanced Networks and Services)
*AT&T*
*BARRNet* (Bay Area Regional Research Network)
*CERFnet* (California Educational and Research Foundation Network)
*CICnet* (Committee on Institutional Cooperation Network)
*Colorado SuperNet*
*CONCERT* (Communication for North Carolina Education, Research and Technology)
*Cornell University*
*CREN* (Corporation for Research and Educational Networking)
*CSUnet* (California State Universities Network)
*JvNCNet*
*MCI Telecommunications, Inc.*
*MichNet / MERIT*
*Midnet*
*MRnet* (Minnesota Regional Network)
*NCAR* (National Center for Atmospheric Research) *
*NEARnet* (New England Academic and Research Network)
*netIllinois* *
*NevadaNet*
*NorthWestNet*
*NYSERNet* (New York State Educational and Research Network)
*OARnet* (Ohio Academic and Research Network)
*Onet* (Ontario Network) *
*PREPnet* (Pennsylvania Research and Economic Partnership Network)
*PSCnet* (Pittsburgh Supercomputer Center Network)
*PSInet* (Performance Systems International Network)
*Sesquinet*
*SDSCNet* (San Diego Supercomputer Center Network)
*SURAnet* (Southeast Universities Research Association Network)
*THEnet* (Texas Higher Education Network)
*VERnet* (Virginia Education and Research Network)
*Westnet*

* = Associate Member

**LAURA BREEDEN** became executive director of FARNET in May 1991. Prior to joining FARNET, she held a variety of positions at Bolt Beranek and Newman Inc. in Cambridge, MA, including CSNET project manager, NEARnet project manager, and manager of the network services group. Ms. Breeden has a B.A. in Urban Studies and Education from Oberlin College, Oberlin, OH, where the computer, in her day, resided in the basement of the physics building.

***There will be a FARNET BOF on Thursday night, May 21 at 7:30pm.***

# Book Reviews

*MIT Project Athena: A Model for Distributed Campus Computing,* by George Champine, Digital Press, 1991, ISBN 1-55558-072-6.

**Background**

In the early days of 1983 the Massachusetts Institute of Technology embarked on an eight year quest to build a distributed computing environment with the goal of improving effectiveness of its educational program. MIT was joined in this challenging journey by numerous sponsors, most notably by Digital Equipment Corporation and International Business Machines, each contributing more than $50 million of equipment, personnel, and cash. The effort was dubbed "Project Athena" after the Greek goddess of wisdom. It was a great success.

Project Athena is interesting for several reasons. First, there is its impact on academic computing and the educational experience at MIT, which was, after all, the major impetus for the project. Then there are the politics and difficulties of putting together a successful $100 million project with sponsorship from two rival companies. Also the immense challenges of leapfrogging an entire generation of computing—moving from mainframe and mini-computer timesharing directly to workstations in a distributed computing environment, skipping personal computers. But, of most interest to readers of *ConneXions* are the problems and solutions devised in building a distributed computing environment on a scale of thousands of machines and tens of thousands of users—and doing so years before the industry had moved beyond target workgroups of tens of machines and tens of users. Add the constraint that personnel support should increase no more than linearly with the number of workstations and that all users should be able to use any Athena workstation anywhere on campus, and you had quite a challenge. One that Athena was successful in meeting.

**Organization**

George Champine's book, *MIT Project Athena: A Model for Distributed Campus Computing,* attempts to tell the Athena story. The book is divided into four major parts: Development, Pedagogy, Technology, and Administration.

*Development* takes one on a whirlwind tour of the first five years of the project, from initial conception to the deployment of the workstation environment.

*Pedagogy* takes a brief look at the educational impact of Project Athena by focusing on the courseware developed for teaching MIT courses, as well as the sometimes bitter lessons learned in the institutional and technical areas of the courseware development process.

*Technology* is the most interesting section for network managers and those who build or run distributed computing environments. It describes the various network services developed or used by Project Athena, spending only a page or two on each service. The services described include the network, name service (*Hesiod*), authentication (*Kerberos*), service management (*Moira*), notification (*Zephyr*), printing (the never deployed *Palladium*), file service (*RVD, NFS,* and *AFS*), electronic mail (*POP* and a centralized mail hub), conferencing (*Discuss*), on line delivery of consulting services (*OLC*), and dialup timesharing (some technologies never die). Champine repeats himself over and over again in this section, describing many of the network services components twice—perhaps because one of its chapters was first published as a standalone article.

*Administration* covers a hodgepodge of subjects including workstations in student residences, finance and organization, and the recent assessments of the project used to shape the future of academic computing at MIT. The discussion of workstations in student living groups and the very brief treatment of Athena's unexpected results are most interesting.

**Good overview**

*MIT Project Athena: A Model for Campus Computing* is a good overview of the Athena quest, the difficulties encountered along the way, and its successes and failures. The subject is large and complex, and the book's biggest problem is that it just touches the surface of everything it covers, appearing superficial in places.

If you are looking for a moderately deep analysis of any aspects of the project, then Champine's book will disappoint you. But if you read it to get an overview of the many aspects of Project Athena and plentiful references to more detailed works, you may enjoy it. But beware—there are numerous inaccuracies. Most are minor technical details, but taken together they could result in a distorted picture.

**A place to start**

The bottom line is if you, like so many of us, can't afford to wait for the solidification vendor vaporware which is supposed to manage your growing distributed computing environment—and that seems several years away, at best—you owe it to yourself to study how Athena solved similar problems, if only so you can figure out how to survive until we all reach the promised land. Champine does a reasonable job at giving you a place to start.

*—Jon Rochlis, MIT Project Athena*

---

*Integrated Broadband Networks—An Introduction to ATM-Based Networks,* by Rainer Händel and Manfred N. Huber, Addison-Wesley, ISBN 0-201-54444-X, 1991, 230 pp.

Like "Asynchronous Transfer Mode—Solution for Broadband ISDN" by Martin de Prycker (ISBN 0-13-053513-3, 1991, 264 pp., Ellis Horwood, reviewed in *ConneXions,* Volume 6, No. 4, April 1992) this is a thorough introduction to B-ISDN and ATM concepts, heavily based on the work in large Telecommunications Equipment companies and PTTs [de Prycker is from Alcatel where Händel and Huber are from Siemens].

**Detailed**

Firstly, let me say that it is less readable than de Prycker, but more detailed. It is very telegraphic in style (and reads more like expanded course notes than a textbook). It has less on specific research and development switch work, but far more on the standards and B-ISDN model itself, as well as on the use of SDH to carry ATM, and would form a good companion.

If you have no feel for ISDN, then I would recommend reading an introductory text such as Stallings' *ISDN and Broadband ISDN* (Macmillan, 1992, ISBN 0-02-415475-X, reviewed in *ConneXions,* Volume 6, No. 4, April 1992), as there are many assumptions made about the readers ability to digest "I.ijk," and other telecommunications jargon.

The book is particularly good on transmission and interworking details, including DQDB MANs, although rather quiet on congestion problems in "Interworking Units" that will glue many of these components together.

*continued on next page*

## Book Reviews *(continued)*

**Organisation**

The book is divided in 10 chapters, as follows:

1. *Introduction:* What is B-ISDN? Where has it come from?
2. *Service Requirements:* Usual list of video/audio/data.
3. *Principles and Building Blocks:* Fixed size cells due to low error rate on optical transmission and high bandwidth requiring cut through switching.
4. *B-ISDN Network Concept:* Virtual Paths and Channels explained.
5. *B-ISDN User Interface and Protocols*
6. *ATM Switching*
7. *ATM Transmission Network*

**Good, clear details**

5, 6 and 7 form the backbone of the technical content of the book. 5 covers B-ISDN equivalent of the OSI behemoth! This includes physical mapping onto SDH in great detail, the ATM layer itself and the various Adaptation Layers (excluding the most recent draft #5). 6 covers switching and is very good in general and the clearest section of the book. 7 covers the transmission and topology of the network.

8. *Evolution Scenarios for B-ISDN:* Outlines the authors' views on the evolutionary path to install services for TV, LAN and MAN interconnection.
9. *Miscellaneous:* (Including connectionless service, voice and echo problems and tariffing!).
10. *Outlook:* The authors are optimistic about the advent of services, and wax eloquent(-ish) on Intelligent Networks (but clearly have not used WAIS/*Xarchie* or network weather services :–). They briefly mention some of the research still required for the subsequent pure optical networks, and finally put a cherry on the icing with a very useful troika of appendices on standardisation, DQDB/ATM interworking and abbreviations.

The book is written in UK English on the evidence of the use of the word "flat" for "apartment."

*—Jon Crowcroft, University College London*

---

*The Little Black Book: Mail Bonding with OSI Directory Services,* by Marshall T. Rose, Prentice Hall, Inc., ISBN 0-13-683210-5, 1992.

Despite the persistent rumors, I don't believe that Marshall Rose is a secret agent sent to destroy OSI. *The Little Black Book* (TLBB) is a strong testament to the possibilities and the pragmatics of making the OSI Directory (X.500) a reality. Rose educates the reader on the topic from a strong hands-on perspective.

**Easy reading**

This is the third book Rose has brought to the industry and *TLBB* exhibits the same consistent excellence as that of his previous works. The style of the book lends it to easy reading despite the technical nature of its content. There is never a dull moment in *TLBB;* it is interesting reading from the first page to the last. In terms of the content, particularly the soapboxes, Rose's personality is unmistakably present and that means the readers are likely to either love or hate the content of the book.

I only have one complaint about *TLBB*. Rose leaves the reader with the clear message that it is time to jump. However, it is not clear which way to jump. This reviewer took away three clear messages from the text; (1) an extensive directory service is critical to the growth of network based services, (2) there is no question that the vision of OSI Directory is compelling, however (3) the pragmatics of making the OSI Directory a reality may simply prevent it from ever fulfilling this vision.

**Soapboxes**

Of course, *TLBB* is loaded with Rose's trademark "soapboxes." Throughout the text Rose gives insight to his personal perspective on controversial issues. Having been involved with OSI Directory in a very hands-on manner, Rose is able to provide many insights into the controversies surrounding the technology *and* the people involved with it. To those already familiar with the technology of OSI Directory the soapboxes provide important insight into the meta-issues surrounding the use of the technology. For readers of Rose's earlier books the soapboxes may seem a bit conservative; however, he has saved the best for last with a soapbox that absorbs the entire final chapter.

**Organization**

*TLBB* begins by laying the high-level groundwork of OSI and the Internet to facilitate the discussion of OSI Directory and make comparisons to the well-known Internet community. Following the overviews Rose dives into the OSI Directory by first providing a high-level overview and then progressing into the gory details of methods to access the OSI Directory service. Each discussion is augmented with practical, real-life, examples drawn from Rose's personal experience with implementing the OSI Directory. The text then turns from the generalities to specifics of the better ways one might access and apply the OSI Directory. This discussion uses specific examples of the *White Pages* pilot run by PSI, use of the OSI Directory to support electronic messaging, interrogation algorithms and various user interface programs. Following discussion of methods to access the OSI Directory, *TLBB* turns to the characteristics of the DSA that occupies the traditional "server" role in the service. Finally, Rose concludes the main body of the text with a final, "all soapbox," chapter giving his insight into what the future holds.

Information about an ongoing OSI Directory project and an actual implementation are provided in the Appendixes to *TLBB*. First it reviews efforts by the NADF to establish an operational directory based on the OSI Directory in North America. Then ISODE is reviewed and specific information on obtaining ISODE is provided.

**Value**

The greatest value in *TLBB* is the practical implementation and usage experience woven throughout the general discussion. This experience truly differentiates *TLBB* from other books covering similar topics. In addition, of a more tactical value, *TLBB* is loaded with excellent illustrations that reduce complex issues into clear diagrams. These diagrams make the topic much more understandable.

**Conclusions**

*The Little Black Book* is mandatory reading for anyone working with the ISODE directory *QUIPU* and/or the *White Pages* pilot effort. It will be of great value to those working with, or considering aspects of, the OSI Directory. If you are an implementor, *TLBB*'s predecessor, *The Open Book,* is an ideal complementary text covering the OSI underpinnings of the Directory. Taken together these texts comprise the best source of information on OSI and OSI Directory available in print. Bottom line: while you may not share all of Rose's perspectives there is no arguing that he is one of the best authors in the industry—regardless of your perspective, *The Little Black Book* is a must.

*—Chris Moore*

# *Opinion:* The Trouble with OSI*

**by Ole J. Jacobsen, Interop Company**

## Introduction

No one can argue against the value of having standards. Go almost anywhere in the world, and you can expect to find film for your camera or batteries for your Walkman. This is a Good Thing. Having standards for computer-communications is also a Good Thing. It allows equipment from different vendors to operate with one another. This is all "Motherhood and Apple Pie," and I'm sure you're going to stop reading if I go any further in explaining the benefits of what has come to be known as *open systems.* I will point out, however, that we actually have such systems today, thanks to TCP/IP, the *de facto* standard for open systems.

## OSI

The largest driving force behind standards for open systems is undoubtedly the *International Organization for Standardization* (ISO) with its *Open Systems Interconnection* (OSI) suite of protocols. ISO, working jointly with the CCITT, gave us the famous 7-layer Reference Model in the early 1980s, and has been hard at work ever since, defining standards for the various parts of OSI. I like to call these pieces the *Components of OSI,* and I've been running a series of articles in this journal over the last four years under that heading. Given the tremendous political and market power behind OSI, its emergence seemed inevitable, and I decided that a set of tutorial articles on this topic would be appropriate. Thus the series.

## Five years

As the articles started to arrive on my desk, I realized what a huge and complex undertaking OSI really is, and I became curious as to when we'd all see the fruits of the OSI labor. The answer varies somewhat depending on who you ask, and depending on what your definition of "real OSI systems" is, but remarkably most people think that five years from now is a good estimate. The problem, however, is that those same people answered the same way five years ago, and I have a strong feeling they will give a similar answer in another five years!

## Why?

It is natural to ask why OSI represents such a sliding target. In this article, I will outline four main causes which I believe add up to a large impediment against OSI:

- The problem area for OSI is much too large
- The standardization process is ineffective
- The standards seem to be in perpetual revision mode
- The road from a standard to stable, market-ready implementations is long and thorny

I will discuss each of these in turn.

*Problem area:* Take almost any OSI standard and you will find a prevailing tendency to solve every conceivable problem. Coupled with this "bells and whistles" philosophy is the problem of *dependency:* most OSI components are intended to operate in conjunction with other parts. Often these other parts are incomplete or under development, a common phrase in ISO/CCITT documents is "For further study." Ultimately, the standards depend on administrative infrastructure. Name registration, for example, requires changes in the bureaucracy of a given target, and the OSI effort hasn't even begun to address such issues, yet.

* *Readers with differing viewpoints are encouraged to respond. Send your letters to:* connexions@interop.com.

*Process:* Perhaps the greatest problem is the standardization process itself. If you gather together computer professionals from all around the world and ask them to agree upon a "common way of doing something" you are almost guaranteed to end up with a suboptimal solution, riddled with options and compromises—and it goes without saying that it will take them a very long time to reach *any* kind of agreement.

*Constant revision cycle:* To make matters worse, the CCITT (and by extension ISO) defines a 4-year "study period" after which a new revision of a standard is issued. The revisions are rarely "backwards compatible," and this makes OSI truly be a moving target.

*From standards to products:* When some part of OSI finally reaches *International Standard* (IS) status, there follows a long period of time in which *implementor's agreements* are defined by so-called *profiling groups.* Unfortunately, there are many regional profiling groups in the world, each defining its own set of agreements and later attempting to "harmonize" the solutions.

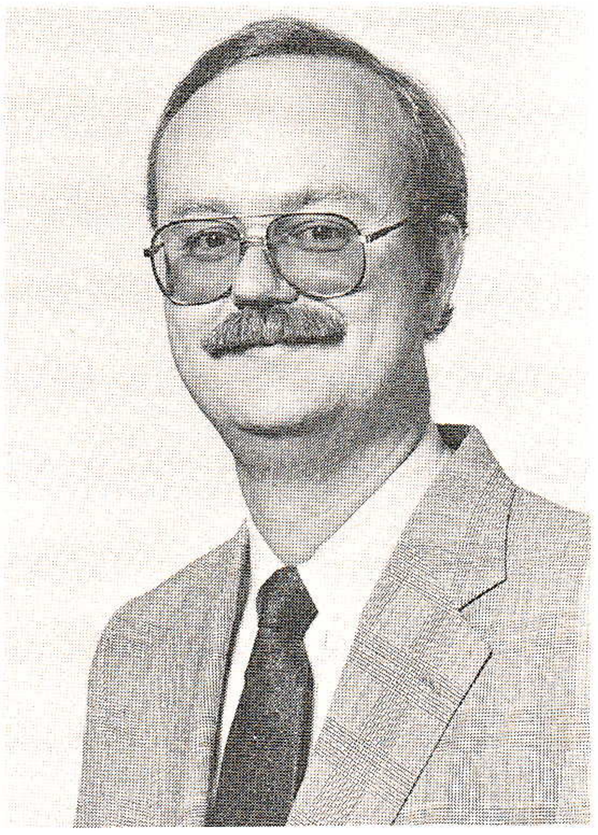
So what to do? When I first heard about OSI, I was given the impression that it would supplant all other forms of computer-communications software, and become as common as those "AA" batteries for the Walkman. On the other side, the people in what is commonly known as "The Internet Community"—those who run TCP/IP—used to say "never in our back yard!" when it came to OSI.

## A change in attitude

Lately, I've started hearing slightly different tunes from both sides. This is quite encouraging. What appears to be happening is two-fold. First, the Internet community has started several OSI trials, having recognized that OSI contains some services that are either missing from, or superior to, the ones in the Internet suite. Second, the "OSI Bigots" now seem far more willing to accept that the future of computer-communications spells "multi-protocol" rather than "all OSI." We should continue to see this convergence, since the standards gurus on both sides are actually now talking to one another. Also, the standardization process is itself becoming more open (in both camps), I've even heard rumors that the hitherto expensive OSI standards documents will soon become more accessible.

## Conclusion

Users and vendors are unlikely to want to "throw out" what they have in order to adopt OSI. In retrospect, had OSI started a few years earlier, limited its focus, and (thus) worked a bit faster, we would perhaps have more "complete OSI" today. Instead, we now enjoy a mix of viable protocols, and we should use these pieces to build our networks for the future.



**OLE-JØRGEN JACOBSEN** has been active in the computer networking field since 1976 when he went to work for the Norwegian Defence Research Establishment, an early ARPANET site. In 1984 he joined the DDN Network Information Center at SRI International. His tasks included user assistance and the compilation of a directory of TCP/IP products. He joined Dan Lynch as employee number three of the newly formed Interop Company. (then called "Advanced Computing Environments") in February 1987, just weeks before the very first INTEROP conference. Since that time, Ole has remained the Editor (and more recently Publisher) of *ConneXions—The Interoperability Report*, a monthly technical journal in the field of computer-communications. He is also the Technical Program Director for the INTEROP conference and tutorials. In his spare time, Ole studies baroque organs and music. He enjoys spending time in the organ loft at Stanford's Memorial Church which houses the magnificent dual-temperament Fisk pipe organ. Ole serves on the Board of Directors for the Palo Alto Chapter of the American Guild of Organists (AGO). Ole also loves telephones and owns his own PBX. His wife is still trying to remember that she has to dial "9" for an outside line. He holds a B. Sc. in Electrical Engineering and Computing Science from the University of Newcastle upon Tyne, England. He can be reached as: `ole@interop.com`.

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

ADDRESS CORRECTION
REQUESTED



CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen